

**UNIVERSIDAD NACIONAL DE PIURA**  
**FACULTAD DE CIENCIAS**  
**ESCUELA PROFESIONAL DE INGENIERIA**  
**ELECTRÓNICA Y TELECOMUNICACIONES**



**TESIS**

**“SISTEMA DE DETECCIÓN Y CONTEO DE VEHÍCULOS  
UTILIZANDO VISIÓN ARTIFICIAL”**

**PRESENTADA POR:**

**PEÑA MERINO, JUIMY MILTON YEFF**

**PARA OPTAR EL TITULO PROFESIONAL DE:  
INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**

**Línea de investigación:**

**Sistemas Digitales**

**Piura, Perú**

**2017**

**UNIVERSIDAD NACIONAL DE PIURA**

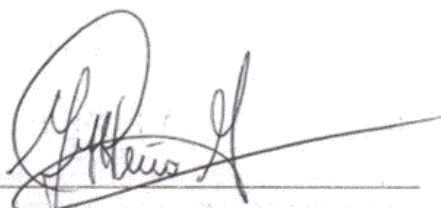
**FACULTAD DE CIENCIAS**

**ESCUELA PROFESIONAL DE INGENIERIA**

**ELECTRÓNICA Y TELECOMUNICACIONES**

**“SISTEMA DE DETECCIÓN Y CONTEO DE VEHÍCULOS  
UTILIZANDO VISIÓN ARTIFICIAL”**

Línea de investigación:  
Sistemas Digitales



Bch. PEÑA MERINO, JUIMY MILTON YEFF  
EJECUTOR



M.Sc. JUAN M. JACINTO SANDOVAL  
ASESOR

## DECLARACIÓN JURADA DE AUTENTICIDAD DE LA TESIS

Yo: Juimy Milton Yeff Peña Merino, identificado con DNI N° 70055323, Bachiller de Escuela Profesional de Ingeniería Electrónica y Telecomunicaciones, de la Facultad de Ciencias y domiciliado en Av. Mario Galán Mz. L15 Lote 57 San Martín del Distrito de Veintiséis de Octubre, Provincia de Piura, Departamento de Piura.

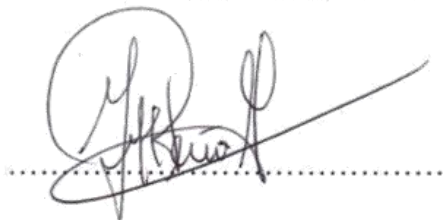
Celular: 945512195

Email: miltonyeff@gmail.com

**DECLARO BAJO JURAMENTO:** que la tesis que presento es auténtica e inédita, no siendo copia parcial ni total de una tesis desarrollada, y/o realizada en el Perú o en el Extranjero, en caso contrario de resultar falsa la información que proporcione, me sujeto a los alcances de lo establecido en el Art. N° 411, del código Penal concordante con el Art. 32° de la Ley N° 27444, y Ley del Procedimiento Administrativo General y las Normas Legales de Protección a los Derechos de Autor.

En fe de lo cual firmo la presente.

Piura octubre del 2017



DNI N° 70055323

Artículo 411.- El que, en un procedimiento administrativo, hace una falsa declaración en relación a hechos o circunstancias que le corresponde probar, violando la presunción de veracidad establecida por ley, será reprimido con pena privativa de libertad no menor de uno ni mayor de cuatro años.

Art. 4. Inciso 4.12 del Reglamento del Registro Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales –RENATI Resolución de Consejo Directivo N° 033-2016-SUNEDU/CD



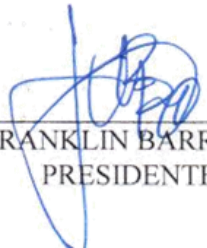
**UNIVERSIDAD NACIONAL DE PIURA  
FACULTAD DE CIENCIAS  
ESCUELA PROFESIONAL DE INGENIERIA**

**ELECTRÓNICA Y TELECOMUNICACIONES**

**“SISTEMA DE DETECCIÓN Y CONTEO DE VEHÍCULOS  
UTILIZANDO VISIÓN ARTIFICIAL”**

Línea de investigación:  
Sistemas Digitales

APROBADA POR:  
JURADO:



---

M.SC. FRANKLIN BARRA ZAPATA  
PRESIDENTE



---

DR. CARLOS ENRIQUE ARELLANO RAMIREZ  
SECRETARIO



---

ING. MARIO AUGUSTO RAMOS ECHEVARRIA  
VOCAL





# UNIVERSIDAD NACIONAL DE PIURA

## FACULTAD DE CIENCIAS



### ACTA DE SUSTENTACIÓN 072-2017-FC-UNP

#### FACULTAD DE CIENCIAS

Los Miembros del Jurado Calificador que suscriben, reunidos para evaluar la Tesis denominada "**SISTEMA DE DETECCIÓN Y CONTEO DE VEHÍCULOS UTILIZANDO VISIÓN ARTIFICIAL**" presentado por el señor Bachiller **PEÑA MERINO – JUIMY MILTON YEFF**, con el asesoramiento del **MSc. Juan Manuel Jacinto Sandoval**; oídas las observaciones y respuestas a las preguntas formuladas, y de conformidad al Reglamento de Tesis para obtener el Título Profesional en la Facultad de Ciencias, lo declaran:

**APROBADO (X)**

**DESAPROBADO ( )**

Con la mención de:

MUY BUENO

(X) En consecuencia, queda en condición de ser ratificado por el Consejo de Facultad de Ciencias de la Universidad Nacional de Piura, y recibir el **TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**.

(X) En consecuencia, queda en condición de ser ratificado por el Consejo Universitario de la Universidad Nacional de Piura, y recibir el **TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO Y TELECOMUNICACIONES**; después que el sustentante incorpore la sugerencia del Jurado Calificador.

Piura, 18 de octubre 2017.

**UNP**

  
M.Sc. FRANKLIN BARRA ZAPATA  
PRESIDENTE DE JURADO DE TESIS

  
Dr. CARLOS ENRIQUE ARELLANO RAMÍREZ  
SECRETARIO DE JURADO DE TESIS

  
Ing. MARIO AUGUSTO RAMOS ECHEVARRIA  
VOCAL DE JURADO DE TESIS



## **DEDICATORIA**

Este gran logro se lo dedico de manera especial a mi familia que ha sido desde el inicio de mis estudios y carrera profesional fuente de gran motivación que han impulsado mis logros y que me han acompañado en todo momento.

Una mención especial a mis padres: Amilton Peña Cordova y María Merino Abad, quienes me apoyan en los momentos más difíciles de mi vida, por sus consejos, sus valores, y porque ellos son el motivo de mi constante superación ya que cada día me dan ánimos para seguir adelante, superándome constantemente tanto en lo personal como en lo profesional.

A mis hermanos Hanghelo y Milagros, por estar siempre conmigo en las buenas y en las malas, por brindarme su gran amor, por darme la fortaleza y el apoyo moral y espiritual, para la culminación del presente trabajo.

## **AGRADECIMIENTO**

Agradecer a Dios, en primer lugar, por su presencia tangible en cada paso que doy, por su orientación y guía al caminar, por darme una familia unida, llena de amor y valores, por las oportunidades con las que me ha permitido toparme, por su conocimiento, por su guarda y por su amor.

A mi familia, por su amor incondicional y gran apoyo en el inicio de cada una de mis aventuras. A mis abuelos, tíos y primos que han contribuido en mi desarrollo tanto personal como profesional.

Agradezco también a mi asesor quién creyó siempre en mi capacidad y me brindó su sapiencia y experiencia en su especialidad como Ingeniero.

A mis amigos, compañeros y futuros colegas de la promoción 2012 de la escuela profesional de Ingeniería Electrónica y Telecomunicaciones, particularmente a: Arturo Ramírez, Carlos Olaya, Cristhian Merino, Raúl Herrera, Richard Espinoza, Josué Saavedra, David Inga, Rubén Inga, Roberto Salvador, Manuel Sosa y William Mendoza. También a mis amigos, de la infancia, del colegio, de la academia y además de personas especiales que me han apoyado y contribuido en cada momento de mi vida.

Finalmente, un agradecimiento especial a mi Alma mater. La “Universidad Nacional de Piura”, a la “Facultad de Ciencias” y a la “Escuela Profesional de Ingeniería Electrónica y Telecomunicaciones”. Y es mi promesa dar mi máximo esfuerzo en mi vida profesional, para no defraudar a esta noble casa de estudios y sembradora de renombrados profesionales.

# ÍNDICE GENERAL

<b>I.</b>	<b>ASPECTOS DE LA PROBLEMÁTICA .....</b>	<b>2</b>
1.1.	DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA.....	2
1.2.	JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN .....	2
1.3.	OBJETIVOS .....	3
1.4.	DELIMITACIÓN DE LA INVESTIGACIÓN.....	3
<b>II.</b>	<b>MARCO TEORICO .....</b>	<b>4</b>
2.1.	ANTECEDENTES DE LA INVESTIGACIÓN.....	4
2.2.	BASES TEÓRICAS.....	6
2.2.1.	CONCEPTOS BÁSICOS DE LA INGENIERÍA DE TRÁFICO.....	6
2.3.	DETECCIÓN DE VEHÍCULOS.....	8
2.4.	MÉTODOS BASADOS EN VISIÓN ESTÉREO .....	9
2.4.1.	Mapas de disparidad .....	10
2.4.2.	Mapeado de perspectiva inversa (Inverse Perspective Mapping, IPM). .....	10
2.4.3.	MÉTODOS BASADOS EN MOVIMIENTO .....	11
2.4.4.	MÉTODOS BASADOS EN APARIENCIA .....	13
2.4.5.	MÉTODOS BASADOS EN EL APRENDIZAJE .....	13
2.4.6.	EXTRACCIÓN DE CARACTERÍSTICAS .....	14
2.4.7.	VISION ARTIFICIAL.....	14
2.4.8.	PROCESAMIENTO DE IMAGENES.....	15
2.4.9.	LA IMAGEN DIGITAL.....	15
2.4.10.	DISPOSITIVOS DE CAPTURA DE IMÁGENES.....	16
2.4.11.	IMÁGENES BLANCO/NEGRO Y COLOR .....	19
2.4.12.	RESOLUCIÓN ESPACIAL Y EN AMPLITUD.....	20
2.4.13.	REPRESENTACIÓN DE IMÁGENES DIGITALES .....	21
2.5.	PROCESAMIENTO Y ANALISIS DE IMÁGENES DIGITALES.....	21
2.6.	PROCESAMIENTO BASICO DE IMÁGENES .....	22
2.6.1.	OPERACIONES INDIVIDUALES .....	23
2.6.2.	OPERADOR IDENTIDAD.....	24
2.6.3.	OPERADOR INVERSO O NEGATIVO .....	24
2.6.4.	OPERADOR UMBRAL.....	25
2.6.5.	TRANSFORMACIÓN DE VECINDAD.....	26
2.7.	OPERACIONES DE FILTRADO .....	28



2.7.1.	<i>Filtros Paso Bajo</i>	28
2.7.2.	<i>Filtros Paso Alto</i>	29
2.8.	HISTOGRAMA	29
2.9.	OPERACIONES MORFOLÓGICAS.	30
2.9.1.	<i>Dilatación y Erosión.</i>	31
2.10.	CONCEPTO DE OPENCV	31
2.10.1.	<i>Características principales</i>	32
2.10.2.	<i>Módulos OpenCV</i>	32
2.11.	GLOSARIO DE TÉRMINOS BÁSICOS	33
2.12.	HIPÓTESIS	36
<b>III.</b>	<b>MARCO METODOLÓGICO</b>	<b>37</b>
3.1.	MÉTODOS Y PROCEDIMIENTOS	37
3.1.1.	<i>ACONDICIONAMIENTO DE LA ESCENA</i>	39
3.1.2.	<i>ADQUISICIÓN DE VIDEO</i>	39
3.1.3.	<i>CONVERSIÓN A ESCALA DE GRISES</i>	39
3.1.4.	<i>FILTRADO PASA BAJO</i>	40
3.1.5.	<i>DIFERENCIA DE IMAGENES</i>	40
3.1.6.	<i>BINARIZADO</i>	40
3.1.7.	<i>DETECCIÓN DE OBJETOS</i>	40
3.1.8.	<i>ETIQUETADO DE OBJETOS</i>	41
3.1.9.	<i>SEGMENTACIÓN</i>	41
3.1.10.	<i>CONTEO</i>	41
<b>IV.</b>	<b>RESULTADOS Y DISCUSIÓN</b>	<b>43</b>
4.1.	RESULTADOS	43
4.2.	DISCUSIÓN	50
	<b>CONCLUSIONES</b>	<b>51</b>
	<b>RECOMENDACIONES</b>	<b>52</b>
	<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>53</b>
	<b>ANEXOS</b>	<b>54</b>

## ÍNDICE DE FIGURAS

FIGURA 1: DIGITALIZACIÓN DE UNA SEÑAL ANALÓGICA .....	17
FIGURA 2: CAPTURA DE UNA IMAGEN 3D POR UN DISPOSITIVO CCD.....	18
FIGURA 3: FIGURA DEL ÁRBOL CAPTURADA POR UNA CÁMARA CON 4X4 SENSORES DE INTENSIDAD .....	19
FIGURA 4: CUATRO REPRESENTACIONES DE LA MISMA IMAGEN CON VARIACIÓN EN EL NÚMERO DE PÍXELES UTILIZADOS .....	20
FIGURA 5: SEIS REPRESENTACIONES DE LA MISMA IMAGEN CON VARIACIÓN EN EL NÚMERO DE NIVELES DE GRIS UTILIZADOS .....	20
FIGURA 6: CONVENCION DE EJES UTILIZADA PARA LA REPRESENTACIÓN DE IMÁGENES DIGITALES .....	21
FIGURA 7: FUNCIONES DE PUNTO Y VECINDAD .....	22
FIGURA 8: OPERACIÓN INDIVIDUAL.....	23
FIGURA 9: REPRESENTACIÓN DEL OPERADOR IDENTIDAD .....	24
FIGURA 10: REPRESENTACIÓN DEL OPERADOR INVERSO .....	25
FIGURA 11: REPRESENTACIÓN DEL OPERADOR UMBRAL .....	26
FIGURA 12: TRES NÚCLEOS REPRESENTATIVOS DE FILTROS PASO BAJO .....	28
FIGURA 13: TRES NÚCLEOS REPRESENTATIVOS DE FILTROS PASO ALTO .....	29
FIGURA 14: HISTOGRAMA.....	30
FIGURA 15 OPERACIÓN MORFOLÓGICA DE UN ELEMENTO ESTRUCTURANTE $S[u,v]$ SOBRE UN ÁREA DE LA IMAGEN $F[x,y]$ PARA OBTENER UNA IMAGEN $G[x,y]$ . .....	30
FIGURA 16: OPENCV .....	32
FIGURA 17 DIAGRAMA DE BLOQUES DE LOS PASOS A SEGUIR .....	38
FIGURA 18 CAPTURA DE IMAGEN PREVIA.....	43
FIGURA 19 CAPTURA DE IMAGEN ACTUAL .....	44
FIGURA 20 IMAGEN PREVIA EN ESCALA DE GRISES.....	45
FIGURA 21 IMAGEN ACTUAL EN ESCALA DE GRISES.....	45
FIGURA 22 IMAGEN PREVIA DESPUES DE APLICAR FILTRO .....	46
FIGURA 23 IMAGEN ACTUAL DESPUES DE APLICAR FILTRO .....	46
FIGURA 24 RESULTADO DE LA DIFERENCIA ENTRE IMAGEN PREVIA Y ACTUAL .....	47
FIGURA 25 IMAGEN BINARIZADA.....	47
FIGURA 26 IMAGEN DESPUES DE APLICAR EROSION Y DILATACIÓN .....	48
FIGURA 27 IMAGEN CON REGIÓN DE VEHÍCULOS DETECTADOS .....	48
FIGURA 28 IMAGEN CON REGIÓN DE VEHÍCULOS: DETECCIÓN Y CONTEO .....	49
FIGURA 29:RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS.....	54
FIGURA 30: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS.....	55
FIGURA 31: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS.....	56
FIGURA 32: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS.....	57

FIGURA 33: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS .....58

FIGURA 34: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS .....59

FIGURA 35 RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS .....60

## ÍNDICE DE ANEXOS

ANEXO 1: RESULTADO DE MAS CAPTURAS DEL SISTEMA .....	54
ANEXO 2: INSTALACIÓN Y CONFIGURACIÓN CON VISUAL STUDIO .....	61
ANEXO 3: CÓDIGO DEL PROGRAMA .....	72

## **RESUMEN**

El presente trabajo ha descrito las técnicas empleadas para realizar un sistema de detección y conteo de vehículos utilizando visión artificial. Para ello se propuso un conjunto de conceptos relacionados al proyecto, desde que es una imagen digital, procesamiento, como transformación a niveles de gris, binarización, morfología matemática, hasta la detección y conteo de vehículos. Este conteo vehicular es uno de los aspectos que componen un sistema de gestión de tráfico, pero es un elemento importante en los procesos de planeación y diseño de la construcción, ampliación y modernización de la infraestructura. Este proyecto presenta un sistema automático de conteo vehicular en tiempo real.

La aplicación obtiene el video de la calle que se requiera analizar a través de una cámara USB o un video que ha sido previamente guardado en un archivo con formato AVI o mp4. Este video se carga en la aplicación para ser procesado y realizar el conteo. La herramienta utilizada para el procesamiento de imágenes es la Librería OPENCV que se configura y utilizan con el lenguaje de programación Visual C++. Para el tratamiento de las imágenes y extraer los datos se utiliza la técnica de sustracción de fondo y primer plano. Ha sido necesaria la creación de funciones que puedan cuantificar el movimiento de los vehículos en el video analizado.

El trabajo incluye una reseña de los métodos más utilizados para la detección de Objetos en una secuencia de imágenes, especialmente en la sustracción de fondo de una escena, que es el componente fundamental para la identificación de objetos en movimiento en este trabajo.

### **Palabras claves:**

Detección de vehículos, conteo de vehículos, visión artificial.

## **ABSTRACT**

The present work has described the techniques used to realize a system of detection and counting of vehicles using artificial vision. To do this, a set of concepts related to the project was proposed, from the existence of a digital image, processing, such as transformation to gray levels, binarization, mathematical morphology, to the detection and content of vehicles. This vehicle counting is one of the aspects that make up a traffic management system, but it is an important element in the planning and design processes of the construction, expansion and modernization of the infrastructure. This project presented an automatic system of counting vehicles in real time.

The application gets the video of the street that is required to analyze through a USB camera or a video that has been previously saved in an AVI or mp4 format file. This video is loaded into the application to be processed and counted. The tool used for image processing is the OPENCV Library that can be used with the Visual C ++ programming language. For the treatment of the images and extraction of the data the technique of subtraction of background and foreground is used. It has been necessary to create functions that can quantify the movement of vehicles in the video analyzed.

The paper includes an overview of the methods most used for the detection of objects in a sequence of images, especially in the background subtraction of a scene, which is the fundamental component for the identification of moving objects in this work.

**Keywords:** Detection of vehicles, counting of vehicles, artificial vision.



## **INTRODUCCIÓN**

La Visión Artificial cada vez adquiere más importancia por sus múltiples aplicaciones a nivel industrial, científico, médico, de seguridad, control de calidad, automatización de procesos obtención, reconocimiento y clasificación de objetos por su tamaño, reconocimiento de rostros, huellas dactilares, entre muchos otros.

Los sistemas de visión artificial utilizan de manera conjunta un amplio número de técnicas diferentes y complementarias, tales como el análisis y procesamiento de imágenes, seguimiento de múltiples objetos deformables, reconocimiento de patrones, etc., nos permite transformar la imagen visual de un objeto en un conjunto de datos digitalizados que pueden ser procesados; la extracción de información de secuencias de imágenes es clave en muchos sistemas de visión artificial para la solución de problemas como la obtención de datos que nos permiten calcular desplazamiento y velocidad de objetos, la modelización tridimensional del entorno, la restauración de imágenes, etc. Es así como el seguimiento y caracterización dinámica de objetos en movimiento permite la utilización de la visión artificial en aplicaciones como el control de tráfico vehicular, pues, nos permite estimar la densidad vehicular el volumen de tráfico y la velocidad que son los tres parámetros principales que describen el flujo vehicular.

El conteo vehicular es uno de los aspectos que componen un sistema de gestión de tráfico, pero es un elemento importante en los procesos de planeación y diseño de la construcción, ampliación y modernización de la infraestructura. Este proyecto presenta un sistema automático de conteo vehicular en tiempo real.

# **I. ASPECTOS DE LA PROBLEMÁTICA**

## **1.1. DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA.**

Los problemas de tráfico, como la movilidad, seguridad y el consumo eficiente, entre otros, han sido siempre una preocupación entre la población. Esto ha generado que en los últimos años aumente el interés por disminuir o eliminar estos problemas, surgiendo así los Sistemas Inteligentes de Transporte (ITS). Lo que se pretende desde la comunidad de investigación ITS es conseguir que los sistemas incorporados en los vehículos (sensores de diferente naturaleza) sean capaces de reconocer el entorno y detectar entre otras cosas, las posibles amenazas que se puedan encontrar, como por ejemplo la posibilidad de un choque.

¿Será posible implementar un sistema de visión artificial para detectar y contar vehículos?

## **1.2. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN**

Las vías rápidas o autopistas son una parte importante del sistema de progreso y desarrollo de un país. La función de las autopistas o carreteras como caminos arteriales entre centros urbanos individuales o ciudades será cada vez más importante para el manejo de flujos de enlace hacia y dentro de las ciudades. Sin embargo, cuando el parque automotor local experimenta un crecimiento acelerado, un problema que puede derivarse es que usualmente la cantidad de policías de tránsito o monitoreo vial es insuficiente para el debido control vehicular. Por otra parte, las empresas privadas dedicadas a la concesión vial necesitan herramientas para elaborar estudios de mantenimiento o negociación de las vías. El conocimiento del flujo vehicular en una intersección permite conocer el grado de ocupación y las condiciones bajo las que ésta funciona. Un análisis de su comportamiento permite definir las tendencias de crecimiento y el momento a partir del cual dejará de prestar un servicio adecuado, convirtiéndose así en un cuello de botella.

Son estos motivos que justifican la elaboración de este proyecto de tesis.

### **1.3. OBJETIVOS**

#### **OBJETIVO GENERAL**

Diseñar un sistema de detección y conteo de vehículos utilizando visión artificial.

#### **OBJETIVOS ESPECÍFICOS**

- Implementar técnicas de procesamiento para mejorar las imágenes.
- Implementar algoritmos de extracción de características.
- Implementar el sistema utilizando archivos de video.
- Implementar el sistema utilizando las librerías de OpenCv.

### **1.4. DELIMITACIÓN DE LA INVESTIGACIÓN**

El proyecto será desarrollado, en condiciones ambientales de iluminación, es decir de día. Se utilizarán videos grabados previamente en el formato avi o mp4. El proyecto se desarrollará en una computadora personal de características básicas.

## **II. MARCO TEORICO**

### **2.1. ANTECEDENTES DE LA INVESTIGACIÓN**

En la Tesis “VERIFICACIÓN DE VEHÍCULOS MEDIANTE TÉCNICAS DE VISIÓN ARTIFICIAL” presentada por Gonzalo Ballesteros Villarreal, en la Universidad Autónoma De Madrid, España, se proponen sistemas de verificación de vehículos mediante métodos basados en aprendizaje. En primer lugar, se realiza un estudio del estado del arte para conocer los problemas actuales en la materia. Después, se muestra la arquitectura de los sistemas que se divide en dos etapas: extracción de características y clasificación. En la primera etapa se realiza una breve exposición de los tipos de características que se van a implementar (simetría, bordes, análisis de componentes principales (PCA) e histogramas de gradientes orientados (HOG)). La etapa de clasificación consiste en una explicación teórica de los clasificadores utilizados en nuestro sistema. Posteriormente, se realiza el desarrollo de estos sistemas, efectuando mejoras para cada uno de ellos. Para el sistema basado en simetría se plantean dos métodos diferentes, introduciéndose una mejora en el segundo método, que consiste en una diferenciación entre ejes compuestos por uno y dos píxeles, junto con una penalización en los valores de simetría para conseguir una mayor diferenciación entre las clases. Respecto al sistema basado en bordes, se utilizan únicamente bordes verticales, donde se analiza el uso de vectores reducidos. Por otra parte, se presenta el uso de la matriz de correlaciones para desarrollar el sistema basado en PCA. En el sistema basado en HOG se estudia qué parámetros son los adecuados para el descriptor en el caso particular de vehículos, proponiéndose descriptores eficientes basados en esta configuración, que pueden ser implementados en sistemas en tiempo real. Finalmente, con los resultados obtenidos en el paso previo se procede a un análisis para los distintos métodos presentando sus principales características y limitaciones.

En la Tesis “DESARROLLO DE UN SISTEMA DE CONTEO Y MONITOREO PARA TRAFICO VEHICULAR” presentada por Rosalía Pérez, en la Universidad Católica Andrés Bello, Venezuela, se proponen el desarrollo de un Sistema de Conteo y Monitoreo para Tráfico Vehicular que realiza el conteo, la clasificación y la detección de la velocidad promedio de los vehículos

que transitan por el área de estudio seleccionada, la Carretera Hoyo de la Puerta-El Placer, que presenta altos índices de congestionamiento. Se seleccionaron dos sensores de tráfico para la realización de dicho sistema, un radar microonda que hace uso del efecto doppler y una cámara que utiliza tecnología de video detección. Este proyecto formará parte de una plataforma de adquisición de datos de tránsito que permitirá la elaboración de estadísticas sobre el volumen de vehículos y reportes de tráfico en tiempo real. Se plantean una serie de objetivos específicos, llevados a cabo a través de una metodología dividida en cinco fases, la cual comprende inicialmente la selección, compra y estudio de los sensores de tráfico, el desarrollo de una aplicación en LabVIEW© para la lectura de datos y la instalación y configuración de los equipos. Como resultado de lo antes descrito se logra comprobar el funcionamiento de los sensores escogidos, a través de una fase de prueba que comprendió la recolección de datos de tres días diferentes, lo que certifica la puesta en marcha del sistema.

En la Tesis presentada por los bachilleres, Jorge E. Faytong Real, Giancarlo J. Moggia Cucalón, de la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL, Facultad de Ingeniería en Electricidad y Computación, denominada “MONITOREO AUTOMÁTICO DE CARRETERAS MEDIANTE EL USO DE UN SISTEMA DE DETECCIÓN, SEGUIMIENTO Y EXTRACCIÓN DE CARACTERÍSTICAS DE VEHÍCULOS CON TÉCNICAS DE VISIÓN POR COMPUTADOR”, hace un seguimiento de vehículos que está compuesto por los submódulos de: Segmentación y Caja englobante. La segmentación se encarga de darle sentido a la información resultante del proceso de detección. Para el procesamiento de estos datos, el módulo de segmentación se basa en la utilización de una librería para la extracción de blobs, desarrollada para OpenCV. Un blob es una estructura compuesta por un conjunto de píxeles adyacentes y sus atributos. Estos conjuntos de píxeles han sido agrupados tomando en cuenta que ellos cumplen con ciertos criterios o parámetros de clasificación y es tomado como un objeto, no como simples píxeles separados y sin sentido. Por otro lado, el módulo de caja englobante se encarga de encerrar a los blobs encontrados para hacer posible la adquisición de datos de entrada necesarios para los componentes de extracción de características básicas y predicción de estados futuros. Es importante mencionar que este módulo es el

encargado de la obtención del centroide de cada blob. El centroide es el centro de la caja englobante del blob. Es en sí la estructura de datos utilizada como entrada principal en los componentes antes mencionados, por lo que su producción es vital.

## **2.2. BASES TEÓRICAS**

### **2.2.1. CONCEPTOS BÁSICOS DE LA INGENIERÍA DE TRÁFICO**

#### **2.2.1.1. *Procedimientos de medida y variables de interés***

Entre los diversos sistemas que existen para la detección de tráfico y que en los siguientes apartados se especificarán, nos servimos de uno que es el más utilizado, para establecer las variables que se usan de manera cotidiana para la obtención de datos sobre los vehículos que circulan por las autopistas. Sobre estas variables destacan una serie de las cuales nos servimos para realizar todo tipo de estudios, hay que comentar que todas estas variables están relacionadas con la franja de tiempo en el que se realiza el análisis, es decir, los datos que se recogen de este sistema, automáticamente se agrupan en periodos de un minuto, con lo cual, todas las variables que a continuación se exponen, cuentan, teniendo en cuenta ese periodo de tiempo.

- Intensidad: Determina el volumen de vehículos que circulan por una autopista
- Velocidad: Marca la velocidad de los vehículos que circulan, esta velocidad es la media ponderada por la intensidad.
- Ocupación: Es el porcentaje de tiempo que un vehículo se encuentra detenido en el detector.
- Composición: Es el porcentaje de vehículos ligeros que han circulado

#### **2.2.1.2. *Intensidad del tráfico (Mapa Continuo)***

Una manera habitual de referirnos al estado del tráfico es consultando los medios informativos donde prevalece el mapa continuo, este mapa que se puede consultar vía web y que a menudo suele aparecer en los informativos, realiza una representación de las zonas “complicadas” por el volumen de tráfico, marca con un nivel de colores según si hay tráfico fluido o denso.



Para el desarrollo de este mapa entran en juego la velocidad máxima de la vía y las variables que hemos mencionado anteriormente.

#### **2.2.1.3. *Flujo Vehicular***

La teoría de flujo del tránsito se basa en el desarrollo de las relaciones matemáticas entre los elementos primarios del flujo vehicular tales como flujo, densidad y velocidad. Mediante el análisis del flujo vehicular se pueden entender las características y el comportamiento del tránsito y se describe la forma como circulan los vehículos en cualquier sistema vial.

Los estudios sobre el flujo vehicular tienen sus comienzos en la década de los 30 con la aplicación de la probabilidad para la descripción de éste, Greenshield realizó estudios modelando el comportamiento de una línea de tráfico, proponiendo una relación lineal entre velocidad y densidad después de la segunda guerra mundial con el aumento en el número de vehículos surgen modelos como Car-Following, teoría de ondas de tráfico y teoría de colas para la descripción del flujo vehicular.

En 1992 los científicos Kai Nagel y Michael Schreckenberg crearon un modelo de flujo de tránsito vehicular con un autómata celular, donde cada célula del autómata equivale ya sea a un vehículo en movimiento con cierta velocidad o a un espacio vacío de la avenida donde se encuentran los vehículos. Algunos de estos modelos fueron desarrollados pensando en la realización de las simulaciones a través de computadoras, el proyecto AISUM es un software comercial para la simulación de tráfico y utiliza como modelo para sus simulaciones al “modelo de Gipps de vehículo siguiente” (Peter Gipps) publicado en 1981.

#### **2.2.1.4. *Clasificación de modelos de Flujo Vehicular***

Debido a la gran variedad de modelos de tráfico vehicular existentes se establecieron algunos criterios para clasificarlos.

- Escala de las variables independientes (continuo discreto o semi-discreto)
- Nivel de detalle (microscópico mesoscópico y macroscópico)
- Representación de los procesos (determinístico, estocástico)

- Funcionamiento (analítica o simulación)
- Escala de aplicación

En general todos los modelos deben ser regidos por las mismas variables fundamentales velocidad, densidad y flujo, y por relaciones o ecuaciones entre dichas variables, similares en su forma.

#### **2.2.1.5. Variables Flujo Vehicular**

**Tasa de flujo (q).** Se define el flujo como la frecuencia o número de vehículos que pasa por un punto o sección transversal dada, de un carril o una calzada, durante un periodo determinado, se expresa como:  $q = n/t$

Donde:

- $q$  = Número de vehículos que pasan por unidad de tiempo (vehículos por periodo)
- $n$  = Número total de vehículos que pasan (vehículos)
- $t$  = periodo determinado (unidad de tiempo)

**Velocidad (v).** La velocidad del tráfico es la distancia recorrida por un vehículo durante una unidad de tiempo regularmente se expresa en kilómetros por hora (km/h).

**La densidad o concentración (k).** Es el número de vehículos que ocupa una longitud específica (d), de un sector de la vía en un momento dado  $k = n/d$

Estas tres variables están relacionadas mediante la ecuación fundamental del flujo vehicular:

$$q \text{ vehículos/tiempo} = v * k$$

### **2.3. DETECCIÓN DE VEHÍCULOS**

Los sistemas de detección de vehículos en carretera ponen su atención en secuencias de vídeo de escenas de tráfico. La finalidad de estos sistemas es la detección de vehículos mediante una cámara de vídeo instalada en el vehículo en el cual nos encontramos. Las imágenes obtenidas por esta cámara son procesadas y se determina si se encuentran vehículos en la imagen y cuál es su

posición. Los problemas más importantes a los que se enfrentan estos sistemas son:

Pérdida de información debido al cambio de secuencias en 3D a imágenes en 2D: el proceso de adquisición de las imágenes implica una transformación del espacio 3D al espacio 2D de la imagen. Este efecto distribuye la información de forma no homogénea en la imagen, por lo que vehículos del mismo tamaño aparecen con diferentes dimensiones en las imágenes según su posición relativa a la cámara.

Movimiento de la cámara: el movimiento de la cámara causa que elementos que son estáticos como puede ser el fondo o la carretera, aparezcan con movimiento en la secuencia. El reto es conseguir eliminar ese movimiento creado por el movimiento de la cámara.

Variación intrínseca del vehículo: aunque los vehículos suelen tener una estructura similar, muchos factores como pueden ser: color, forma, tamaño, entre otros, presentan una alta variabilidad.

Cambio en los patrones de apariencia: a parte de los factores comentados anteriormente, la apariencia tanto del fondo como la del vehículo pueden verse seriamente afectados por la variabilidad del medio, sobre todo cuando se producen cambios en las condiciones climáticas o en la iluminación.

Procesamiento en tiempo real: estos sistemas requieren que la detección de vehículos se realice en tiempo real. Para lograr este objetivo se tiene que considerar el tiempo de cómputo de los algoritmos utilizados.

## **2.4. MÉTODOS BASADOS EN VISIÓN ESTÉREO**

En el proceso de obtención de la escena para la detección de vehículos se produce la pérdida de la tridimensionalidad cuando usamos métodos que se basan en visión monocular. Estos métodos, por lo tanto, buscan la obtención de esa tercera dimensión para, en la medida de lo posible, obtener la forma de los objetos. Los sistemas estereoscópicos utilizan al menos dos imágenes distintas de la escena para obtener la información de profundidad. A partir del contenido de estas imágenes se puede determinar la distancia a la que se encuentra el

objeto del observador. La captura de estas imágenes se puede realizar con una o más cámaras. No obstante, típicamente se realiza mediante sistemas compuestos por dos cámaras separadas horizontalmente, simulando el sistema humano de percepción. Otro aspecto que destacar en los sistemas de visión estereoscópica es su geometría, donde encontramos geometrías con los ejes ópticos paralelos o convergentes. El sistema visual humano se basa principalmente en ejes convergentes, donde los ojos enfocan al objeto que queremos visualizar. Por otro lado, los sistemas de visión estereoscópica artificial suelen utilizar ejes paralelos debido a la simplicidad que introducen en el cómputo frente a geometrías convergentes.

#### **2.4.1. Mapas de disparidad**

El conjunto de diferencias que encontramos entre los píxeles de las imágenes de una misma escena se conoce como mapas de disparidad. Si los parámetros de la plataforma del sistema estéreo son conocidos, un mapa en 3D de la escena puede ser creado a partir de los mapas de disparidad. La generación de hipótesis de obstáculos se hace generalmente mediante la búsqueda de grupos de columnas de píxeles con disparidad similar o mediante la explotación de la información de profundidad de alguna otra manera. Los mapas de disparidad se han utilizado en muchos trabajos sobre detección de vehículos. Por ejemplo, los mapas de disparidad, junto con el flujo óptico, son utilizados como una señal visual, y cada píxel de la imagen se clasifica como objeto en movimiento o fondo. El rendimiento óptimo de clasificación se obtiene con un perceptrón multicapa de 3 capas. Como es de esperar, el cálculo de la disparidad no obtiene resultados positivos para todos los píxeles de la imagen. En la literatura se pueden encontrar varios métodos para superar este problema.

#### **2.4.2. Mapeado de perspectiva inversa (Inverse Perspective Mapping, IPM).**

En cualquier proceso de adquisición de imágenes existe un efecto de perspectiva inherente debido a la ubicación de la cámara. Esto no corresponde con una inversión real de mapeo de perspectiva, el cuál es matemáticamente

imposible. Más bien, denota una inversión en virtud de la restricción adicional de que los puntos mapeados inversamente se encuentran en el plano horizontal. Podemos encontrar una explicación de la geometría del mapeado de perspectiva. Si se considera un punto  $p$  en el espacio 3D, el mapeado de perspectiva implica una línea que pasa por este punto y el centro de la proyección  $N$ . Para encontrar la imagen del punto, intersecamos la línea con el plano de la imagen. IPM se define mediante el siguiente procedimiento.

Para un punto  $p_0$   $I$  en la imagen, se traza el rayo asociado a través de  $N$  hacia el plano horizontal. La intersección del rayo con el plano horizontal es el resultado de la asignación de perspectiva inversa aplicada al punto de la imagen  $p_0$   $I$ . Si componemos tanto la perspectiva como la perspectiva inversa, el plano horizontal se asigna a sí mismo, mientras que las partes elevadas de la escena aparecen distorsionadas.

### **2.4.3. Métodos basados en movimiento**

A diferencia de otros métodos que buscan la separación entre los objetos que queremos determinar, en nuestro caso los vehículos y el fondo, los métodos basados en movimiento determinan los vehículos potenciales buscando el movimiento de los objetos en la escena. La aproximación principal que encontramos para este tipo de métodos es la obtención de movimiento mediante el cálculo del flujo óptico, el cual crea un mapa con los vectores de movimiento de todos los píxeles (flujo óptico denso) o de un conjunto de elementos (flujo óptico disperso). En particular, los vehículos que se mueven más rápido que el observador (es decir, los adelantamientos y salidas de vehículos) producen un flujo óptico convergente en el punto de fuga, mientras que los vehículos más lentos producen una divergencia respecto a el flujo óptico.

El flujo óptico se estima a partir de derivadas espaciotemporales de las imágenes de valores de gris con un enfoque local. El flujo óptico estimado se agrupa para eliminar valores atípicos. Suponiendo una cámara calibrada y conocido el ego-movimiento se detectan ambos tipos de objetos, en movimiento y estacionarios. Sin embargo, el flujo óptico denso supone un cómputo lento, por lo tanto, el uso del flujo óptico disperso generalmente se prefiere para

aplicaciones en tiempo real. Este es el caso, por ejemplo, donde el flujo óptico se calcula sólo en los ejes relevantes de la imagen. Para la estimación de flujo óptico se utiliza el método de gradiente, llamado VIGM (Virtual Intensity Gradient Method), en el que se asume un gradiente de intensidad lineal alrededor de los bordes. El flujo óptico sólo se calcula en posiciones concretas, obteniéndose de la siguiente manera: una fracción de las posiciones se determina mediante el uso del algoritmo de muestreo conocido como remuestreo secuencial Monte Carlo, mientras que la fracción restante de las posiciones se determina mediante el uso de una variable aleatoria.

Sin embargo, el movimiento de la cámara que se produce al estar instalada en el vehículo supone un reto en la interpretación del flujo óptico. De hecho, los vectores de flujo óptico reflejan el movimiento de los elementos de imagen en relación con el observador. Por lo tanto, ya que la cámara se está moviendo, los píxeles pertenecientes al fondo proporcionan vectores de flujo óptico que no son cero, incluso si son estáticos, que además difieren de un píxel a otro debido al efecto de la perspectiva. Algunos métodos intentan inferir el ego-movimiento, de modo que ambas fuentes de movimiento pueden ser disociadas. Por ejemplo, estiman el ego-movimiento mediante el cálculo de la matriz esencial entre dos vistas consecutivas de una escena utilizando correspondencias de características. El método explota el conocimiento del ego-movimiento, junto con la extracción de los parámetros del plano de la carretera, para calcular de manera explícita la homografía entre los planos de tráfico en los dos puntos de vista, de manera que las imágenes pueden ser alineados y los objetos en movimiento detectados. Sin embargo, la estimación del ego-movimiento no es una tarea fácil, sobre todo si las imágenes se componen de objetos en movimiento, como es el caso. En efecto, para derivar la matriz esencial sólo las características pertenecientes a elementos estáticos se deben tener en cuenta. Por lo tanto, para extraerlo, las regiones de la imagen que contiene los vehículos tienen que ser conocidas de antemano o inferidas, de modo que las características dentro de ellos se quedan fuera para el cálculo de la matriz esencial.



#### **2.4.4. Métodos basados en apariencia**

La apariencia de los vehículos ha sido siempre una línea importante de estudio cuando se trata de la detección de vehículos. Estos métodos buscan a través del conocimiento de la apariencia de los vehículos, extraer de la escena las regiones donde se encuentran. Los métodos que nos podemos encontrar se pueden agrupar a su vez en dos categorías: métodos basados en modelos y métodos basados en el aprendizaje. Ambas clases se basan en la extracción de las características que podemos encontrarnos en la apariencia de los vehículos. Los métodos basados en modelos usan esta extracción de característica para establecer una correlación entre la imagen y unas plantillas preestablecidas relativas a la apariencia de los vehículos. Por otro lado, los métodos basados en el aprendizaje tratan la detección de vehículos como un problema de clasificación entre dos clases: la clase vehículo y la clase no vehículo. En estos métodos, a priori se realiza el entrenamiento de un clasificador mediante las características extraídas de un conjunto de imágenes representativo de la variabilidad de las clases. Luego, para cada imagen que queramos determinar si se corresponde con un vehículo o no, se realiza la extracción de esas características y se clasifica mediante el clasificador entrenado. Los métodos basados en aprendizaje de la necesidad de definir un método que capturase la variabilidad de los vehículos de manera eficiente, algo muy difícil de obtener mediante los métodos basados en modelos, ya que sería necesario una gran base de datos con todas las posibles plantillas a utilizar y esto resulta inviable para aplicaciones en tiempo real.

#### **2.4.5. Métodos basados en el aprendizaje**

Mediante este tipo de métodos, en el caso de los vehículos, nos encontramos con un problema de clasificación de dos clases (vehículos, no-vehículos). La primera etapa que se realiza en estos métodos es una etapa de aprendizaje: a partir de un conjunto de imágenes, a las que llamaremos conjunto de entrenamiento, se realiza el entrenamiento de un clasificador mediante el uso de las características extraídas previamente de la imagen. El conjunto de características extraídas de la imagen constituye un vector representativo de dicha imagen. Es importante que

el conjunto de entrenamiento tenga una alta variabilidad para conseguir un clasificador robusto. Para determinar que imágenes son vehículos, se extrae un vector de características y se procede a su clasificación. Hay trabajos que, utilizan este tipo de métodos para la detección de vehículos. En algunos proyectos se hace uso de diferentes tipos de características relativas a la apariencia como son las sombras producidas por los vehículos, la simetría inherente de estos o las esquinas que forman la estructura de los vehículos para entrenar el clasificador y realizar la detección de vehículos en la escena. Otros trabajos, utilizan otro tipo de características como son los histogramas de gradientes orientados para realizar la detección.

#### **2.4.6. Extracción de características**

La elección de un conjunto de características adecuado resulta realmente importante para la obtención de un buen rendimiento en la detección. Las características que podemos extraer a partir de los vehículos se pueden clasificar como, características explícitas y características implícitas. Las primeras implican un conocimiento previo de cómo es el objeto, en nuestro caso los vehículos, por lo que hacen uso de sus atributos para definir el descriptor. Por otro lado, las características implícitas no hacen ninguna suposición acerca de la forma, estructura o apariencia de los objetos que se van a clasificar.

#### **2.4.7. VISION ARTIFICIAL**

También conocida como visión por computador (del inglés computer vision) o visión técnica, es un subcampo de la inteligencia artificial y es el campo de acción más ambicioso del procesamiento digital de imágenes. Básicamente el objetivo es automatizar funciones de inspección visual, tradicionalmente utilizadas por el hombre como, por ejemplo.

Los objetivos típicos de la visión artificial incluyen:

La detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes (por ejemplo, caras humanas).

La evaluación de los resultados (ej.: segmentación, registro).

Registro de diferentes imágenes de una misma escena u objeto, hacer concordar un mismo objeto en diversas imágenes.

Seguimiento de un objeto en una secuencia de imágenes.

Mapeo de una escena para generar un modelo tridimensional de la escena; tal modelo podría ser usado por un robot para navegar por la escena.

Estimación de las posturas tridimensionales de humanos.

Búsqueda de imágenes digitales por su contenido.

#### 2.4.8. PROCESAMIENTO DE IMAGENES<sup>1</sup>

**Imagen:** Es la proyección en perspectiva en el plano bidimensional de una escena tridimensional en un determinado instante de tiempo  $t_0$ .

**Fotograma:** Es una matriz bidimensional de valores de intensidad lumínica obtenidos para un tiempo  $t_0$  constante. Pudiera decirse en cierta forma que es una imagen discretizada.

**Píxel** (Picture Element): Es cada de una de las posiciones en que es discretizada una imagen, o lo que es lo mismo, cada una de las posiciones de un cuadro.

**Imagen binaria:** Son aquellas imágenes cuyos píxeles solo tienen dos valores: cero y uno.

#### 2.4.9. LA IMAGEN DIGITAL<sup>2</sup>

Son el principal ingrediente de lo que se conoce como Visión Artificial y representan mediante algún tipo de codificación, normalmente en una matriz de números de dos dimensiones, una escena del entorno.

Existen dos tipos de imágenes utilizadas frecuentemente en Visión Artificial: imágenes de intensidad e imágenes de alcance (también llamadas imágenes de profundidad o perfiles de superficie). Las imágenes de intensidad

---

<sup>1</sup> Rafael C. Gonzales, Richard E. Woods, “Tratamiento Digital de Imágenes”

<sup>2</sup> Gonzalo Pajares Martinsanz, Jesús Manuel de la Cruz García, José Manuel molina pascual, Juan Cuadrado Pardo, Alejandro López Correa, “Imágenes digitales - Procesamiento practico con java”

miden la cantidad de luz que incide en un dispositivo fotosensible, mientras que las imágenes de alcance estiman directamente la estructura en tres dimensiones (3D) de la escena ya que su fundamento radica en el uso de sensores de alcance ópticos y algún fenómeno físico para adquirir la imagen. Un ejemplo típico de una imagen de intensidad es una fotografía, mientras que de una imagen de alcance es, por ejemplo, la imagen que obtiene el oftalmólogo sobre el grado de rugosidad de la córnea de un paciente o las imágenes de un radar.

Aunque la filosofía de los diferentes tipos de imágenes es diferente, en cualquier caso, tras su captura tendremos una matriz de valores en dos dimensiones (2D), es decir, una imagen digital.

#### **2.4.10. DISPOSITIVOS DE CAPTURA DE IMÁGENES**

Para la adquisición de imágenes digitales se requieren dos elementos básicos. El primero es un dispositivo físico que es sensible a una determinada banda del espectro de energía electromagnético (tal como rayos X, ultravioleta, visible, infrarrojo, etc.) y que produce una señal eléctrica de salida proporcional al nivel de energía incidente en cualquier instante de tiempo. El segundo, denominado digitalizador, es un dispositivo que cumple la función de convertir la señal eléctrica continua de salida del dispositivo físico en un conjunto discreto de localizaciones del plano de la imagen y, después, en la cuantización de dicha muestra. Esto implica, en primer lugar, determinar el valor de la imagen continua en cada una de las diferentes localizaciones discretas de la imagen (cada valor localizado de forma discreta se denomina muestra de la imagen) y, luego, asignar a cada muestra una etiqueta entera discreta, que es representativa del rango en el que varía la muestra.

Una vez capturada la señal continua y cuantificada espacialmente y en amplitud, se obtiene una imagen digital, que es como se representa en el computador, es decir, tendremos la matriz (2D) de números como ya hemos mencionado anteriormente. Éstos son los valores que se manipulan para extraer información de las imágenes mediante programas (software).

En la siguiente figura se ilustra un ejemplo de la cuantización espacial y en amplitud:

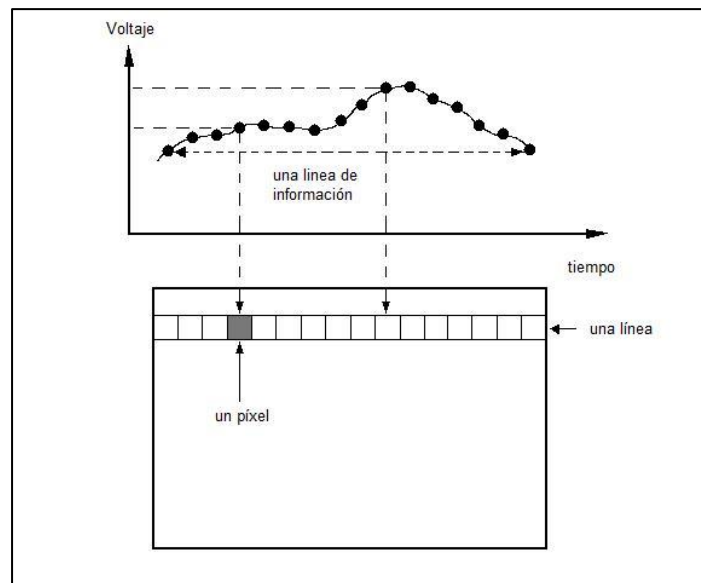


FIGURA 1: DIGITALIZACIÓN DE UNA SEÑAL ANALÓGICA

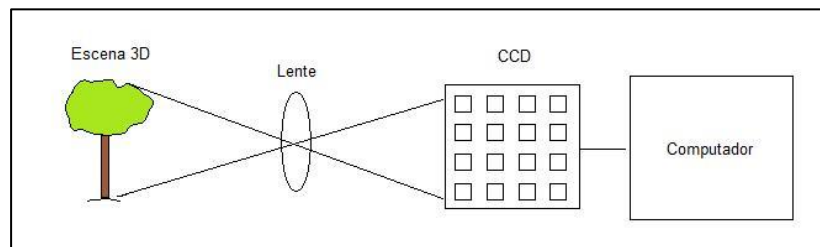
Supongamos que tenemos una señal analógica, que bien podría ser una línea de video analógica. Esta señal (línea) analógica de video se convierte a una imagen digital muestreando la señal analógica a intervalos determinados. El procedimiento consiste en medir el voltaje de la señal a intervalos de tiempo fijos. El valor del voltaje en cada instante se convierte a un número que es almacenado y se corresponde con la intensidad de la imagen en ese punto. La intensidad en cada punto depende tanto de las propiedades intrínsecas del objeto que se está viendo como de las condiciones de luz de la escena. Repitiendo este procedimiento para todas las líneas de video que constituyen una imagen, se pueden grabar los resultados obtenidos en el computador, de suerte que habremos conseguido una imagen digital que, en definitiva, es una matriz de números.

La imagen puede accederse como una matriz bidimensional (2D) de datos, donde cada punto o dato se denomina píxel.

Además de las cámaras de televisión que generan una imagen de video, uno de los sensores más usados para la visión artificial son los dispositivos de acoplamiento de carga (Charge Coupled Devices - CCD). Entre los dispositivos CCD, que generalmente también producen una señal continua de video, cabe distinguir dos categorías: sensores de exploración de línea y sensores de

exploración de área. Estos sensores CCD se basan en unos elementos semiconductores llamados photosites. Los fotones procedentes de la escena excitan el elemento semiconductor, de forma que el grado de excitación es proporcional a la cantidad de carga acumulada en el photosite y, por lo tanto, a la intensidad luminosa en ese punto.

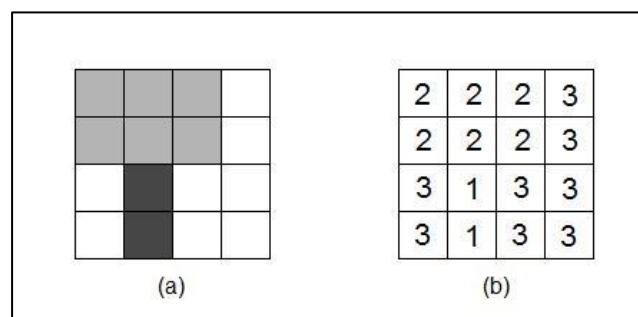
Estos photosites se pueden representar en forma de matriz como se muestra en la siguiente figura:



*FIGURA 2: CAPTURA DE UNA IMAGEN 3D POR UN DISPOSITIVO CCD*

Supongamos una matriz de photosites situados detrás de una lente y sobre los que se proyecta una imagen procedente de la escena 3D. La señal de estos sensores se procesa en el propio sensor (cámara) o en otro dispositivo (tarjeta de procesamiento de imágenes digitales) y los valores digitales se envían al computador.

Para clarificar un poco más estos conceptos, analicemos el ejemplo sencillo mostrado en la figura siguiente:





Supongamos una escena 3D cuya representación digital en forma de matriz, tal como se almacena en el computador, resulta ser la que se muestra en la figura 3-(b). En este caso se utiliza una matriz de 16 elementos (4x4) para representar el árbol; lógicamente, cuantos más sensores dispongamos en la cámara, más precisión podemos tener en la reproducción de la imagen. Esto es lo que se conoce como resolución espacial de la imagen.

#### 2.4.11. IMÁGENES BLANCO/NEGRO Y COLOR

En definitiva, e independientemente del tipo de sensor utilizado, la imagen que ha de ser tratada por el computador se presenta digitalizada espacialmente en forma de matriz con una resolución de  $M \times N$  elementos.

Si la imagen es en Blanco y Negro (B/N), se almacena un valor por cada píxel. Se suele utilizar un rango de valores para su representación, que generalmente es de 0 a  $2^n - 1$ . Uno de los valores más utilizados de  $n$  es 8; esto significa que el rango de valores para este caso varía de 0 a 255. En este caso, el 0 representa el negro absoluto y el 255, el blanco absoluto. Esto indica que podemos tener una resolución o precisión en los grises posibles de 256. El hecho de utilizar 256 niveles es porque con 8 bits del computador se pueden codificar 256 valores distintos desde la combinación 00000000, que representa el nivel 0, hasta la combinación 11111111, que representa el nivel 255.

En el caso de las imágenes en color, los elementos de la matriz vienen dados por tres valores, que representan cada uno de los componentes básicos del color en cuestión. Estos componentes son el Rojo (R), Verde (G) y Azul (B), el conocido código RGB. En este caso el conjunto de valores (0,0,0) es el negro absoluto; el (255,255,255), el blanco absoluto; el (255,0,0), el rojo puro; el (0,255,0), el verde puro; el (0,0,255), el azul puro. Como es lógico, la combinación de distintos valores proporciona otros colores, por ejemplo, el (255, 255, 51) es un tono de amarillo o el (204, 153, 102), es un tono marrón. El número de colores posible resulta ser 255.

#### 2.4.12. RESOLUCIÓN ESPACIAL Y EN AMPLITUD

Para comprender mejor el sentido y la diferencia que existe entre resolución espacial y en amplitud, vamos a ilustrar los conceptos en las siguientes figuras:

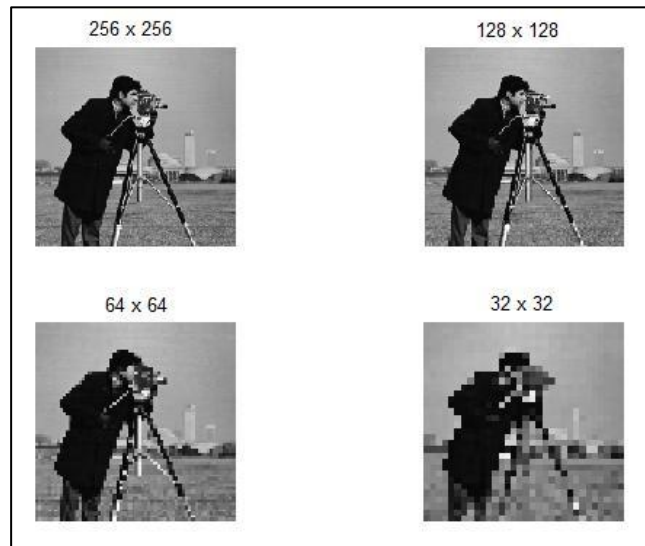


FIGURA 4: CUATRO REPRESENTACIONES DE LA MISMA IMAGEN CON VARIACIÓN EN EL NÚMERO DE PÍXELES UTILIZADOS

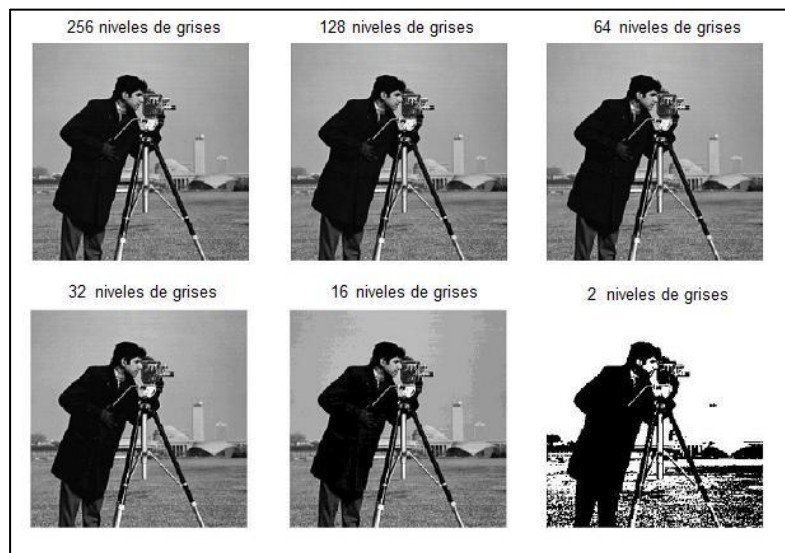
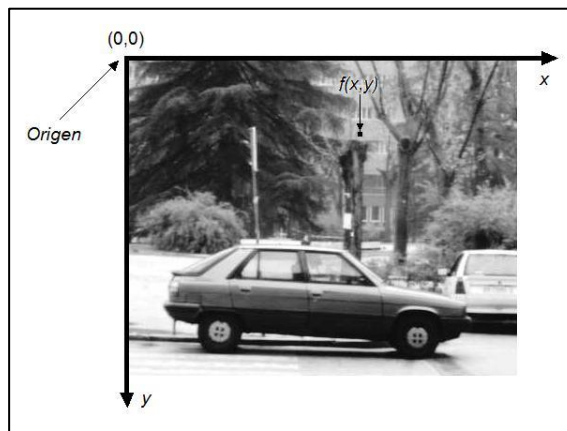


FIGURA 5: SEIS REPRESENTACIONES DE LA MISMA IMAGEN CON VARIACIÓN EN EL NÚMERO DE NIVELES DE GRIS UTILIZADOS

Como se muestra en las figuras anteriores, dependiendo del número de píxeles que tenga el dispositivo y de niveles de grises con que se trabaje en el computador, la imagen poseerá más o menos resolución espacial y en amplitud respectivamente.

#### **2.4.13. REPRESENTACIÓN DE IMÁGENES DIGITALES**

Como ya se ha mencionado antes, el termino imagen se refiere a una función de intensidad bidimensional, la cual puede ser representada como  $f(x,y)$ ,  $f(i,j)$ ,  $I(x,y)$ ,  $I(i,j)$ , etc., donde  $x$  e  $y$ , o bien,  $i$  y  $j$  son las coordenadas espaciales y el valor de  $f$  o  $I$  en cualquier punto  $(x,y)$  o  $(i,j)$  es proporcional a la intensidad o nivel de gris de la imagen en ese punto.



*FIGURA 6: CONVENCION DE EJES UTILIZADA PARA LA REPRESENTACION DE IMAGENES DIGITALES*

#### **2.5. PROCESAMIENTO Y ANALISIS DE IMÁGENES DIGITALES**

Aunque la distinción entre procesamiento y análisis de imágenes digitales no es obvia de forma inmediata, el procesamiento de imágenes puede ser visto como una transformación de una imagen a otra imagen, es decir, a partir de una

imagen, se obtiene otra imagen modificada. Por otro lado, el análisis es una transformación de una imagen en algo distinto a una imagen; en consecuencia, el análisis es un determinado tipo de información representando una descripción o una decisión. En la mayoría de los casos, las técnicas de análisis de imágenes digitales son aplicadas a imágenes que han sido procesadas previamente. Desde el punto de vista de un observador humano, el análisis de imágenes es una tarea fácil y rápida, algo que no ocurre en visión artificial. Igualmente, la capacidad de percepción humana permite procesar rápidamente una imagen para detectar en ella características de interés, por ejemplo, bordes o regiones.

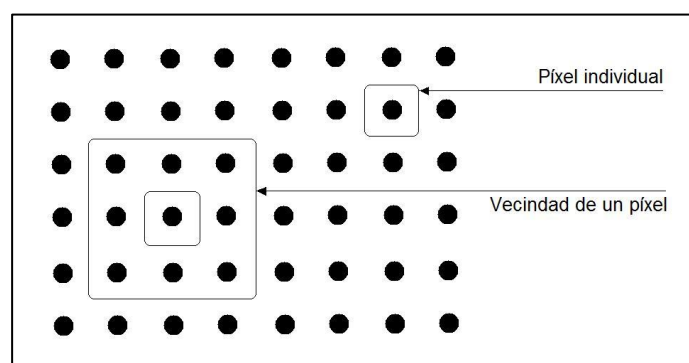
## 2.6. PROCESAMIENTO BASICO DE IMÁGENES

El procesamiento de datos en el sistema de visión puede enfocarse desde 2 perspectivas:

Alteración píxel a píxel de los datos en una escala global (individuales).

Operaciones basadas en múltiples puntos (vecindad).

La generación de un nuevo píxel en una imagen será una función bien del valor de cada píxel en su localización individual, o bien de los valores de los píxeles en la vecindad de un píxel dado, como se indica en la figura siguiente:



*FIGURA 7: FUNCIONES DE PUNTO Y VECINDAD*

Existen aún otras operaciones, que no se clasifican ni como individuales ni como vecindad ya que transforman las imágenes por otros procedimientos; son

transformaciones que operan globalmente sobre los valores de intensidad de la imagen cuyo efecto es un realzado de la imagen original, operaciones aritméticas y lógicas, estas últimas basadas en la teoría del algebra de Boole, y operaciones que realizan transformaciones geométricas, sin modificar los valores de intensidad.

### 2.6.1. OPERACIONES INDIVIDUALES

Las operaciones individuales implican la generación de una nueva imagen modificando el valor del pixel en una simple localización basándose en una regla global aplicada a cada localización de la imagen original. El proceso consiste en obtener el valor del píxel de una localización dada en la imagen, modificándolo por una operación lineal o no lineal y colocando el valor del nuevo pixel en la correspondiente localización de la imagen nueva. El proceso se repite para todas y cada de las localizaciones de los pixeles en la imagen original.

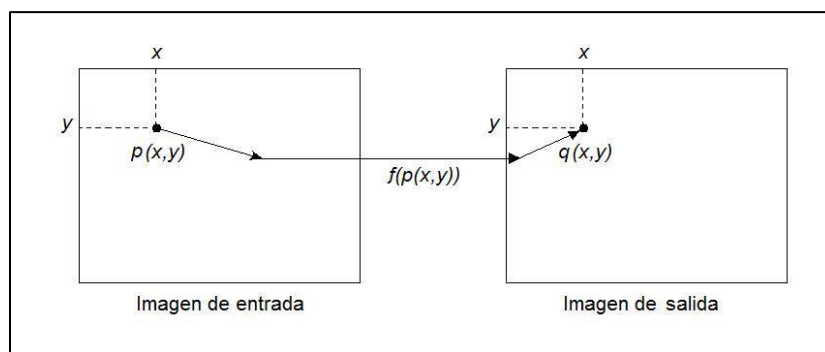


FIGURA 8: OPERACIÓN INDIVIDUAL

Como se aprecia en la figura anterior, el operador individual es una transformación uno a uno. El operador  $f$  se aplica a cada píxel en la imagen o sección de la imagen y la salida depende únicamente de la magnitud del correspondiente píxel de entrada; la salida es independiente de los píxeles adyacentes. La función transforma el valor del nivel de gris de cada píxel en la imagen y el nuevo valor se obtiene a través de la ecuación:

$$q(x,y) = f(p(x,y))$$

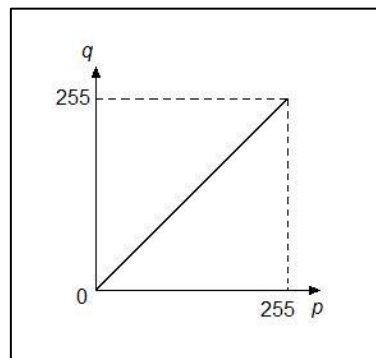
La función  $f$  puede ser un operador lineal o no lineal. El proceso matemático es relativamente simple. La imagen resultante es de la misma dimensión que la original.

### 2.6.2. OPERADOR IDENTIDAD

Este operador crea una imagen de salida que es idéntica a la imagen de entrada. La función de transformación es:

$$q=p$$

El operador identidad deja la imagen de entrada invariante. En la siguiente figura se muestra la función de transformación dada por la ecuación anterior:



*FIGURA 9: REPRESENTACIÓN DEL OPERADOR IDENTIDAD*

### 2.6.3. OPERADOR INVERSO O NEGATIVO

Este operador crea una imagen de salida que es inversa de la imagen de entrada. Este operador es útil en diversas aplicaciones tales como imágenes

médicas. Para una imagen con valores de gris en el rango de 0 a 255 la función de transformación resulta ser:

$$q = 255 - p$$

En la siguiente figura se muestra la función de transformación dada por la ecuación anterior:

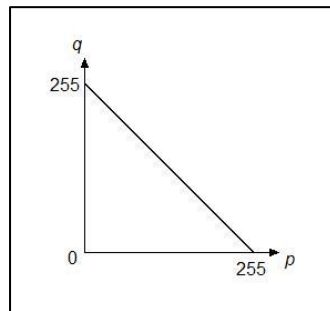


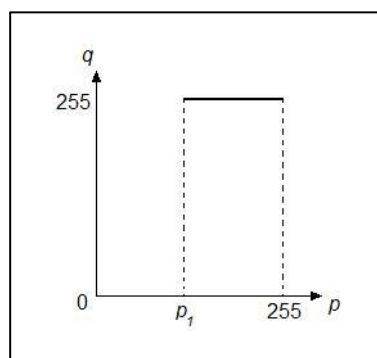
FIGURA 10: REPRESENTACIÓN DEL OPERADOR INVERSO

#### 2.6.4. OPERADOR UMBRAL

Esta clase de transformación crea una imagen de salida binaria a partir de una imagen de grises, donde el nivel de transición está dado por el parámetro de entrada  $p_1$ . La función de transformación es la siguiente:

$$q = \begin{cases} 0 & \text{para } p \leq p_1 \\ 255 & \text{para } p > p_1 \end{cases}$$

En la siguiente figura se muestra la función de transformación dada por la ecuación anterior:



## 2.6.5. TRANSFORMACIÓN DE VECINDAD

En las operaciones de vecindad modifican el nuevo valor del píxel en la imagen de salida depende de una combinación de los valores de los píxeles en la vecindad de la imagen original que está siendo transformada.

Dentro de la categoría de operaciones de vecindad se incluyen las operaciones de filtrado. Las operaciones de filtrado tienen la particularidad de eliminar un determinado rango de frecuencias de las imágenes.

### 2.6.5.1. *Nociones y propiedades de vecindad*

Se dice que todo píxel  $p$ , de coordenadas  $(x,y)$ , tiene cuatro píxeles que establecen con él una relación de vecindad horizontal o vertical, que son:

Horizontal:  $(x-1, y)$  y  $(x+1, y)$  Vertical:  $(x, y-1)$  y  $(x, y+1)$

Estos cuatro píxeles definen lo que se conoce como entorno de vecindad-4 y nos referimos a ellos como  $E_4(p)$ .

Los cuatro vecinos diagonales de  $p$  tienen coordenadas:

$$(x-1, y-1), (x+1, y-1), (x-1, y+1), (x+1, y+1)$$

y nos referimos a ellos como  $E_D(p)$ . Estos píxeles junto con los  $E_4(p)$  se llaman los vecinos-8 de  $p$  y se denotan como  $E_8(p)$ .

Existen excepciones dadas cuando el píxel  $(x, y)$  es un punto del borde de la imagen, en cuyo caso algunos de los vecinos definidos anteriormente no existen.



### 2.6.5.2. *Conectividad*

Sea  $V$  el conjunto de valores de intensidad de los píxeles que se permiten estén adyacentes, por ejemplo, si sólo se desea que exista conectividad entre los píxeles con intensidades 80,81 y 83, entonces  $V = \{80,81,83\}$ . Consideremos tres tipos básicos de conectividad:

**Conectividad-4.** Dos píxeles  $p$  y  $q$  con valores de  $V$  están 4-conectados si  $q$  está en el conjunto  $E_4(p)$ .

**Conectividad-8.** Dos píxeles  $p$  y  $q$  con valores de  $V$  están 8-conectados si  $q$  está en el conjunto  $E_8(p)$ .

**Conectividad-m** (mixta). Dos píxeles  $p$  y  $q$  con valores de  $V$  están m-conectados si  $q$  está en el conjunto  $E_4(p)$  o  $q$  está en  $E_D(p)$  y  $E_4(p) \cap E_4(q) = \emptyset$ .

Un píxel  $p$  es contiguo a otro píxel  $q$  si están conectados. Se puede definir la adyacencia-4, 8 o m, dependiendo del tipo de conectividad especificada. Dos subconjuntos imagen  $S1$  y  $S2$  son contiguos si algún píxel de  $S1$  es contiguo a algún píxel de  $S2$ .

Un camino desde el píxel  $p$  con coordenadas  $(x, y)$  hasta un píxel  $q$  con coordenadas  $(s, t)$  es una secuencia de varios píxeles con coordenadas,

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$

donde  $(x_0, y_0) = (x, y)$  y  $(s, t), (x_i, y_i)$  son adyacentes a  $(x_{i-1}, y_{i-1})$ , con  $1 \leq i \leq n$ , y  $n$  es la longitud del camino. Se pueden definir caminos -4, 8 o m dependiendo del tipo de adyacencia usada.

Si  $p$  y  $q$  son píxeles de un subconjunto de imagen  $S$ , entonces  $p$  está conectado a  $q$  en  $S$  si existe un camino desde  $p$  hasta  $q$  formado de píxeles pertenecientes a  $S$ . Dado un píxel  $p$  cualquiera de  $S$ , el conjunto de píxeles de  $S$  que están conectados a  $p$  se llaman componente conectado de  $S$ . Se deduce

que dos píxeles cualesquiera de un componente conectado están a su vez conectados entre sí y que los componentes conectados distintos son disjuntos.

Un camino simple es un camino sin píxeles repetidos y un camino cerrado es un camino simple en el cual el primer píxel es un vecino del último.

## 2.7. OPERACIONES DE FILTRADO

Las operaciones de filtrado basan su operatividad en la Convolución de la imagen utilizando el denominado núcleo de Convolución. Cabe distinguir dos tipos de filtros: paso alto y paso bajo, que en el contexto de la teoría de señales supone que los primeros dejan pasar las altas frecuencias de la señal y los segundos, las bajas. En el caso de las imágenes nos referimos a frecuencias espaciales. De una forma, las altas frecuencias se asocian a cambios bruscos de intensidad en pequeños intervalos espaciales, es decir, bordes, mientras que las bajas frecuencias se refieren a cambios lentos en la intensidad.

### 2.7.1. Filtros Paso Bajo

En la siguiente figura se muestran algunos núcleos de convolución que caracterizan los filtros paso bajo:

$$PB_1 \equiv \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$PB_2 \equiv \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$PB_3 \equiv \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

FIGURA 12: TRES NÚCLEOS REPRESENTATIVOS DE FILTROS PASO BAJO

### 2.7.2. Filtros Paso Alto

En la siguiente figura se muestran algunos núcleos de Convolución que caracterizan los filtros paso alto:

$$PA_1 \equiv \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$PA_2 \equiv \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$PA_3 \equiv \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

*FIGURA 13: TRES NÚCLEOS REPRESENTATIVOS DE FILTROS PASO ALTO*

## 2.8. HISTOGRAMA

El histograma de una imagen es la representación gráfica o analítica de la distribución relativa de cada valor posible de pixel de imagen, y en caso de imágenes grises de 8 bits será un vector de 256 componentes, siendo la componente  $i$  el número de pixeles de nivel  $i$  en la imagen, dividido por el número total de pixeles:

El histograma es formalmente la función estadística de densidad de probabilidad en forma discreta de los distintos niveles de gris dentro de la imagen.

$$histograma[i] = \frac{N_{\text{pixeles de nivel } i}}{N_{\text{total}}} \quad 0 \leq i \leq 255$$

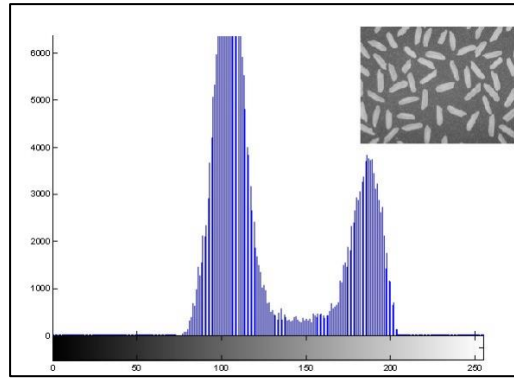


FIGURA 14: HISTOGRAMA

## 2.9. OPERACIONES MORFOLÓGICAS.

Las operaciones morfológicas a imágenes se definen como procedimientos en los cuales cada nuevo pixel de la imagen resultante es obtenido de una operación no lineal entre un conjunto de puntos de la imagen original  $F[x, y]$  y un conjunto de puntos conocido con el nombre de elemento estructurante  $S[x, y]$ . Este elemento estructurante recorre toda la imagen para obtener todos los puntos de la nueva imagen, sin cambiar el tamaño de la imagen de salida.

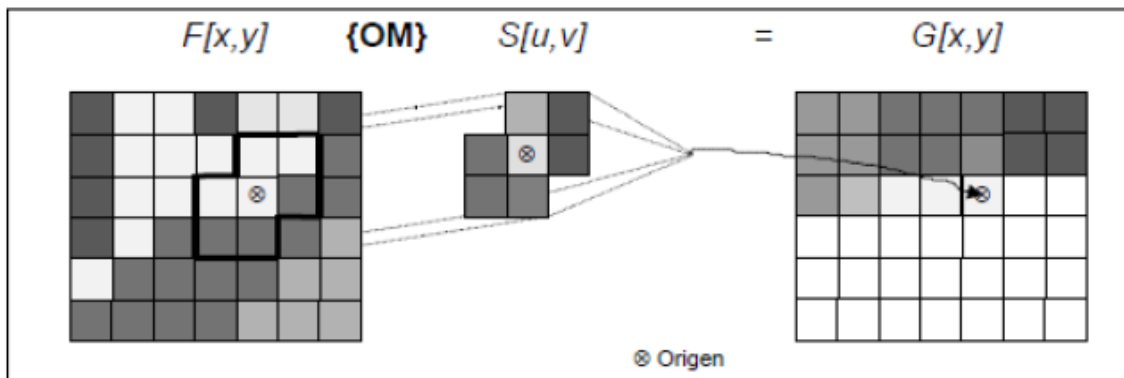


FIGURA 15 Operación morfológica de un elemento estructurante  $S[u,v]$  sobre un área de la imagen  $F[x,y]$  para obtener una imagen  $G[x,y]$ .

Dependiendo de los elementos estructurantes y de las operaciones utilizadas, los filtros morfológicos, pueden detectar bordes en las imágenes, filtrar objetos de tamaños menores a uno determinado, suavizar fondos de texturizados, detectar fallos en patrones de textura, etc.

### **2.9.1. Dilatación y Erosión.**

Las operaciones morfológicas binarias de dilatación y erosión para imágenes binarias consisten en tomar el elemento estructurante (uno de cuyos puntos se considera el origen) y superponerlo a la imagen.

Si en el proceso de erosión el elemento estructurante está completamente contenido en la imagen, el punto de ésta se mantiene, de lo contrario pasa a ser del color del fondo. La erosión se aplica para reducir salientes de objetos, eliminar partes de tamaño menor que el elemento estructurante, eliminar conexiones insignificantes entre objetos mayores.

En el proceso de dilatación si parte del elemento estructurante es igual al contenido de la

imagen, el punto de la imagen pasa a ser igual al punto origen del elemento estructurante, de otra forma pasa a ser del color del fondo. La dilatación se aplica para rellenar entrantes de objetos y unir pequeñas separaciones en las imágenes.

## **2.10. CONCEPTO DE OPENCV<sup>3</sup>**

OpenCV es una biblioteca multiplataforma que permite desarrollar aplicaciones de visión por ordenador en tiempo real. Se centra en el procesamiento de imágenes, captura de vídeo y análisis, incluyendo características como detección de rostros y detección de objetos.

OpenCV significa Open Source Computer Vision Library; por lo tanto, es una librería de tratamiento de imágenes, destinada principalmente a aplicaciones de visión por computador en tiempo real. Una de las ventajas principales es que puede funcionar en muchas plataformas, existen versiones para Windows, Linux y MacOS.

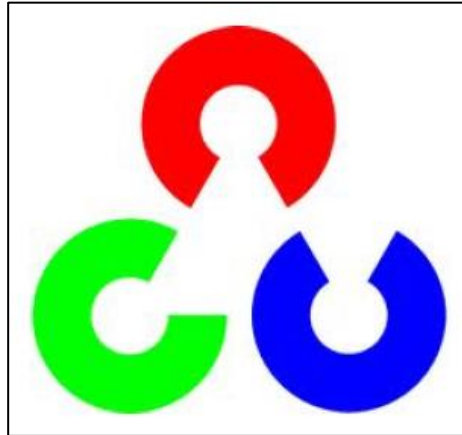
OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel; desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo

---

<sup>3</sup> <https://www.tutorialspoint.com/opencv/>

licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

OpenCV es multiplataforma, existiendo versiones para GNU/Linux, Mac OSX y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.



*FIGURA 16: OPENCV*

#### **2.10.1. Características principales**

- Optimizado para procesamiento de imágenes en tiempo real y aplicaciones de visión artificial
- La interfaz primaria de OpenCV está en C++
- También hay interfaces C, Python y JAVA completas
- Las aplicaciones OpenCV se ejecutan en Windows, Android, Linux, Mac e iOS
- Optimizado para procesadores Intel

#### **2.10.2. Módulos OpenCV**

OpenCV tiene una estructura modular. A continuación, se enumeran los módulos principales de OpenCV.

##### **Núcleo**

Este es el módulo básico de OpenCV. Incluye estructuras de datos básicas (por ejemplo, estructura de datos Mat) y funciones básicas de procesamiento de

imágenes. Este módulo también es ampliamente utilizado por otros módulos como highgui, etc.

### **Highgui**

Este módulo ofrece funciones sencillas de interfaz de usuario, varios códecs de imagen y vídeo, capacidades de captura de imagen y video, manipulación de ventanas de imagen, manejo de barras de seguimiento y eventos de ratón y etc. Si desea capacidades de interfaz de usuario avanzadas, debe utilizar marcos de interfaz de usuario como Qt, WinForms, etc, por ejemplo, cargar y mostrar imágenes, capturar vídeo desde un archivo o cámara, escribir imagen y vídeo en el archivo

### **Imgproc**

Este módulo incluye algoritmos básicos de procesamiento de imágenes incluyendo filtrado de imágenes, transformaciones de imagen, conversiones de espacio de color y etc.

### **Vídeo**

Este es un módulo de análisis de video que incluye algoritmos de seguimiento de objetos, algoritmos de substracción de fondo y etc.

### **Objdetect**

Esto incluye algoritmos de detección y reconocimiento de objetos para objetos estándar.

OpenCV ahora se utiliza extensivamente para desarrollar el procesamiento avanzado de la imagen y las aplicaciones de la visión de la computadora. Ha sido una herramienta para estudiantes, ingenieros e investigadores en todos los rincones del mundo.

## **2.11. GLOSARIO DE TÉRMINOS BÁSICOS**

**ANÁLISIS DEL MOVIMIENTO:** Consiste en comparar una serie de imágenes secuenciales y comprobar que existe una diferencia entre la posición u

localización de un objeto dado, de esta manera se comprueba que este se está moviendo, ubicando así el lugar a donde se mueve y la posición exacta en que se está moviendo. Como sostiene Nalwa (1993) la variación del movimiento relativo entre los objetos estacionarios entre sí y en relación con las diferentes distancias a las que se encuentra respecto al observador que se mueve a su vez respecto de los objetos. Es cuanto se ha movido un objeto de un punto a otro.

**TRANSFORMACIONES LÓGICAS:** consisten en aplicar operaciones lógicas como and, or, xor, etc. A la imagen transformada en un mapa de bits binarios, de esta manera es como aplicarle las operaciones lógicas sobre 0, o 1 lógicos, (equivalente a v o f), con un mapa de bits a diferentes escalas de grises, obteniendo así nuevas imágenes con detalles más realzados.

**CLASIFICACIÓN DE IMAGEN:** La clasificación de imagen hace referencia a la tarea de extraer clases de información de una imagen. El ráster resultante de la clasificación de imagen se puede utilizar para crear mapas temáticos. Dependiendo de la interacción entre el analista y el equipo durante la clasificación, existen dos tipos de clasificación: supervisada y no supervisada.

**SISTEMA:** Un sistema es un conjunto ordenado de elementos que se encuentran interrelacionados entre sí para lograr un fin en común.

**CLASIFICACIÓN SUPERVISADA:** La clasificación supervisada utiliza firmas espectrales obtenidas de las muestras de capacitación para clasificar una imagen. Con la ayuda de la barra de herramientas Clasificación de imagen, puede crear fácilmente muestras de capacitación para representar las clases que desea extraer. También puede crear con facilidad un archivo de firma a partir de las muestras de capacitación que, a continuación, las herramientas de clasificación multivariante utilizarán para clasificar la imagen.

**LOCALIZACIÓN DE OBJETOS EN IMÁGENES:** Cuando queremos localizar un objeto en una imagen, una técnica utilizada consiste en recorrer toda la imagen aplicando múltiples ventanas detectoras de diferentes tamaños. Para cada ventana se aplica algún clasificador que nos indica si existe el objeto que intentamos. Al ser un algoritmo que se basa en diversos recorridos de la imagen, el desplazamiento de la ventana a través de ella es un factor muy importante. Con la disminución del desplazamiento de la ventana, el tiempo de proceso crece



exponencialmente. Por otro lado, un desplazamiento demasiado grande podría comportar la no detección de objetos por no quedar localizados dentro de su ventana.

**BLOB ANÁLISIS:** Un algoritmo utilizado en visión artificial que identifica objetos segmentados y los mide según parámetros morfométricos (tamaño, diámetro, perímetro, etc.) o densitométricos (nivel de gris medio, media, color)

**ATRIBUTO:** Un atributo es una especificación que define una propiedad de un Objeto, elemento o archivo. También puede referirse o establecer el valor específico para una instancia determinada de los mismos.

**CCD:** es la abreviatura de Charge-Coupled Device. Un sensor CCD es un dispositivo semiconductor sensible a la luz, que convierte las partículas de luz (fotones) en cargas eléctricas (electrones). Las cámaras CCD son una de los dos tipos de cámaras que dominan el mercado de la visión, conjuntamente con las cámaras CMOS.

**CMOS:** Es el acrónimo de Complementary Metal Oxide Semiconductor. Esta tecnología funciona como un fotodiodo donde la luz genera una corriente que es representativo de la cantidad de luz que impacta en cada píxel, por tanto difiere significativamente de la tecnología CCD. Hay un número de ventajas en la utilización de los sensores CMOS con respecto a los CCD entre los que se destaca el coste, velocidad, anti-blooming, y características de respuesta programable (Ej. Respuesta con múltiples pendientes).

**ESCALA DE GRISES:** En visión este término se refiere a una imagen monocroma con una gradación de grises. Una cámara monocroma que trabaje a 8 bits generará una imagen con 256 tonos de gris. Si la cámara trabaja a 12 bits su escala de grises será de 4096 niveles.

**HISTOGRAMA:** Es una representación gráfica de los valores de todos los píxeles de la imagen. Generalmente a la izquierda de la gráfica se representa el valor cero correspondiente al negro y a la derecha se representa el valor máximo (que depende del número de bits) correspondiente al blanco. La curva del histograma representa la cantidad de píxeles de cada valor de gris que hay en la imagen. Si la imagen es en color se acostumbran a representar los histogramas de valores de rojo, verde y azul.

**MODELO:** Un modelo es una representación de un objeto, sistema o idea, de forma diferente al de la entidad misma. El propósito de los modelos es ayudarnos a explicar, entender o mejorar un sistema. Un modelo de un objeto puede ser una réplica exacta de éste o una abstracción de las propiedades dominantes del objeto.

**INTELIGENCIA ARTIFICIAL:** Es aquella inteligencia exhibida por artefactos creados por humanos (es decir, artificial). A menudo se aplica hipotéticamente a los computadores. El nombre también se usa para referirse al campo de la investigación científica que intenta acercarse a la creación de tales sistemas.

**ROBOTICA:** Es la rama de la tecnología que se dedica al diseño, construcción, operación, disposición estructural, manufactura y aplicación de los robots. La robótica combina diversas disciplinas como son: la mecánica, la electrónica, la informática, la inteligencia artificial, la ingeniería de control y la física. Otras áreas importantes en robótica son el álgebra, los autómatas programables, la animatrónica y las máquinas de estados.

**MORFOLOGÍA MATEMÁTICA:** es una teoría y técnica para el análisis y tratamiento de las estructuras geométricas, basada en la teoría de conjuntos, teoría de retículos, topología y funciones aleatorias. La morfología matemática es comúnmente aplicada más a las imágenes digitales, pero puede ser empleada también en gráficos, mallas poligonales, sólidos y muchas otras estructuras espaciales.

## **2.12. HIPÓTESIS**

Utilizando visión artificial se detectará y se contará el número de vehículos presentes en las carreteras

### **III. MARCO METODOLÓGICO**

#### **3.1. MÉTODOS Y PROCEDIMIENTOS**

Para el desarrollo del trabajo de tesis se hizo uso del siguiente material:

- 1 computadora personal de 1.2 GHz, 4GB RAM, 500GB DD.
- Una webcam de 29 fps, resolución de 1280 x 740 pixeles.
- Sistema operativo Windows 8.1.

Como lenguaje de programación se utilizó Microsoft Visual C.Net. Además, se hizo uso de las librerías de Visión Artificial, OpenCV, estas librerías son software libre y además que se pueden utilizar en C++, tienen una funcionalidad y flexibilidad fácil de utilizar. Por otro lado, Visual C es un potente lenguaje de programación que da velocidad de procesamiento.

Para la utilización de estas librerías, hay que bajar los archivos de su página principal de OpenCV, luego hay que configurarlas para poder utilizarlas, en todo caso hay un anexo de cómo hacerlo al final del documento de tesis.

Vamos a tener en cuenta que para aplicar el algoritmo se hará uso de dos imágenes. Para eso se memorizará la imagen actual capturada en una variable, para que, en la siguiente captura y proceso, sea utilizada en la detección.

A continuación, se presenta el diagrama de bloques general para llevar a cabo el objetivo planteado y luego se detalla cada uno de los pasos:

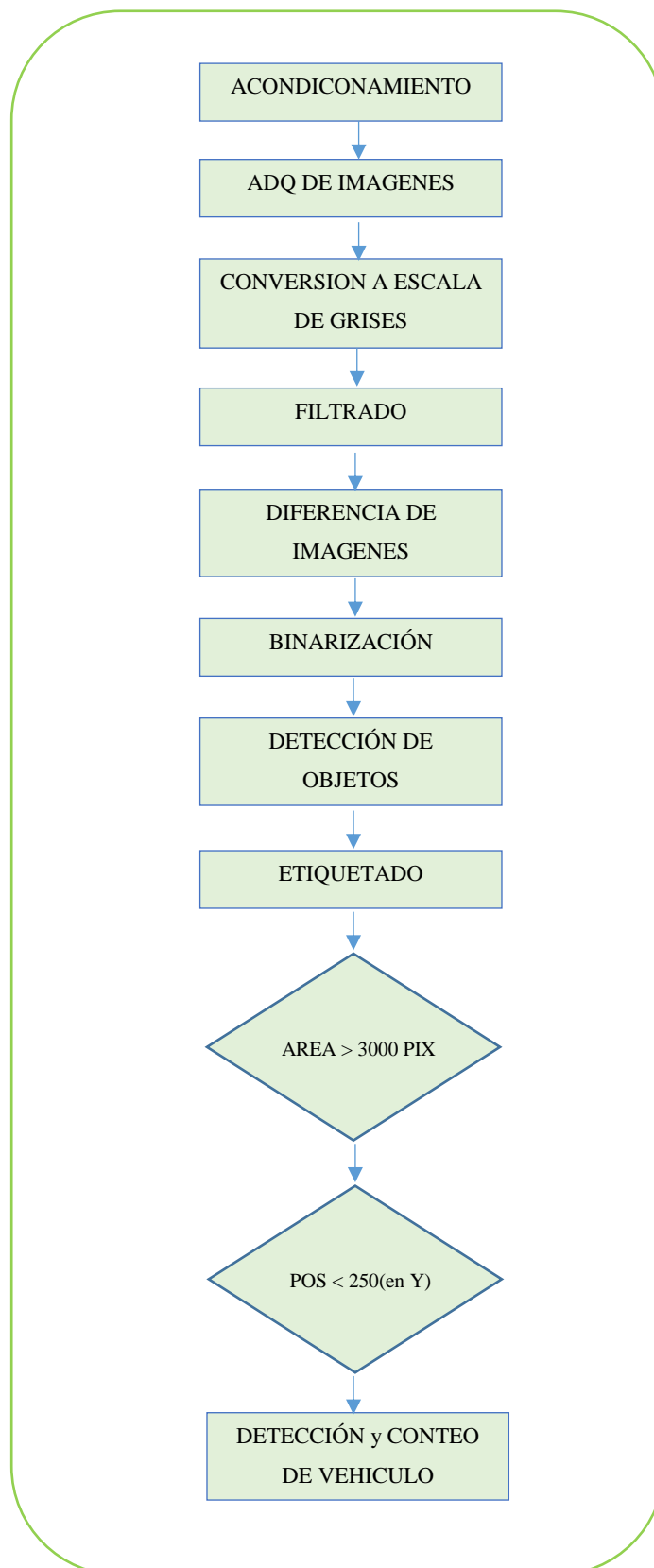


FIGURA 17 DIAGRAMA DE BLOQUES DE LOS PASOS A SEGUIR

### **3.1.1. ACONDICIONAMIENTO DE LA ESCENA**

Todo sistema de visión artificial se lleva a cabo acondicionando la escena, pero una aplicación o proyecto como este, se puede iluminar con reflectores led, o teniendo en cuenta la luz infrarroja. Pero como mayormente el tráfico es en el día vamos a tener en cuenta las condiciones de iluminación del día.

La cámara debe colocarse en un poste para poder visualizar los vehículos desde arriba, casi perpendicular a la pista, con un cierto grado de inclinación, pero que no sea demasiado, ya que los vehículos se verían juntos y el sistema no podría diferenciarlos de forma individual.

### **3.1.2. ADQUISICIÓN DE VIDEO**

La toma de las secuencias de video, se hacen en el formato RGB. Para el propósito de este proyecto se hicieron grabaciones y a partir de ellas, se trabajó con secuencias de video previamente grabado. Las imágenes son de tamaño 1280 pixeles de ancho, por 740 pixeles de alto, que nos daría una resolución de 1 Megapíxel, aunque no se necesita muy buena resolución ya que el sistema solo debe detectar la presencia o no de vehículos.

La velocidad de los fotogramas con que se trabajo fue de 29 fotogramas por segundo.

Para el método que se emplea se procesarán 2 imágenes consecutivas, la actual y la memorizada anteriormente.

### **3.1.3. CONVERSIÓN A ESCALA DE GRISES**

Para el procesamiento posterior y teniendo en cuenta la finalidad, que es la detección y luego el conteo, hay que pasar por convertir las imágenes en escala de grises, es decir, no es necesario trabajar con 3 componentes por pixel, solo una. Por lo tanto, los valores obtenidos van desde 0 a 255.

#### **3.1.4. FILTRADO PASA BAJO**

El filtrado pasa bajo se hace con la finalidad de eliminar ruido presente por la cámara misma. Se utiliza un filtro promediador, cuyo kernel es  $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ . Este kernel o núcleo se desliza por toda la imagen, simplemente toma el promedio de todos los píxeles en el área del núcleo y reemplaza el elemento central., teniendo como resultado un suavizado de la imagen.

#### **3.1.5. DIFERENCIA DE IMAGENES**

Para detectar los vehículos, que son los objetos de interés, se emplea el método de diferencia de la imagen actual y la imagen capturada anteriormente.

Como los vehículos se encuentran en movimiento y la cámara fija, entonces al restar las imágenes, van a resaltar las zonas o bordes de los objetos. Todos los objetos fijos, pasan a formar parte del fondo, como es la pista, veredas, postes, etc.

#### **3.1.6. BINARIZADO**

El siguiente paso será resaltar esos bordes detectados en la diferencia de imágenes. A los objetos que son los vehículos les asignamos el color blanco y al fondo le asignamos color negro. Este proceso de quedarnos con una imagen en blanco y negro se llama binarización. El umbral que hemos tenido en cuenta es de 30, píxeles con valores menores a este umbral le asignamos 0 ó negro y píxeles mayores a este valor le asignamos 255, o color blanco. El valor del umbral 30 fue obtenido mediante pruebas. La función que se utiliza es threshold.

#### **3.1.7. DETECCIÓN DE OBJETOS**

Para detectar completamente los objetos, necesitamos rellenar las regiones obtenidas de los pasos anteriores. Es decir, los bordes, que en el paso anterior teníamos los objetos de color blanco necesitan ser rellanados para obtener un objeto completo. Aquí hacemos uso de la morfología matemática, con dilataciones y erosiones. Una primera erosión que me permite separar o eliminar

objetos muy pequeños que no necesitamos, por ejemplo, las líneas de las pistas, que aparecen y desaparecen por oclusión de los vehículos. Y luego 6 dilataciones que permiten engrosar y juntar los bordes cercanos obtenidos, o sea, un píxel cercano de los objetos.

### **3.1.8. ETIQUETADO DE OBJETOS**

Para separar cada objeto vehículo de la imagen, hay que etiquetarlos. La forma de etiquetar cada objeto número es mediante una función de las librerías `opencv`, `findContours`. Esta función me devuelve una identificación individual de píxeles vecinos y conectados de cada objeto vehículo de la imagen. Además, devuelve el área de cada objeto vehículo de la imagen, como sabemos el área de cada objeto está dado por el número de píxeles presente en ese objeto.

### **3.1.9. SEGMENTACIÓN**

Ahora que tenemos etiquetados cada objeto de la imagen y además su área, podemos filtrar el tamaño de cada objeto para la siguiente etapa. Podemos decir que píxeles mayores a un mínimo y píxeles menores a un máximo van a ser reconocidos. Con esto evitamos que píxeles de objetos pequeños, producidos por el ruido, no entren a la etapa de reconocimiento. Se está considerando vehículo a los objetos que contenga más de 3000 píxeles.

Cada vehículo encontrado en las imágenes es enmarcado con un rectángulo, además, del número del vehículo. En la parte superior, se especifica la cantidad total.

### **3.1.10. CONTEO**

La etapa del Conteo depende de la dirección. Para el caso específico, estamos haciendo un conteo de vehículos que van hacia arriba, por lo tanto, la esquina superior izquierda de la imagen está la coordenada 0,0 y en la esquina inferior derecha está la coordenada 1280,740.

Hacemos una línea imaginaria entre los puntos 0, 250 y 1280, 250. Objetos que toquen y pasen hacia arriba esta línea, los contabilizamos y comenzamos a incrementar de uno en uno.

Los resultados se pueden apreciar en el siguiente capítulo.

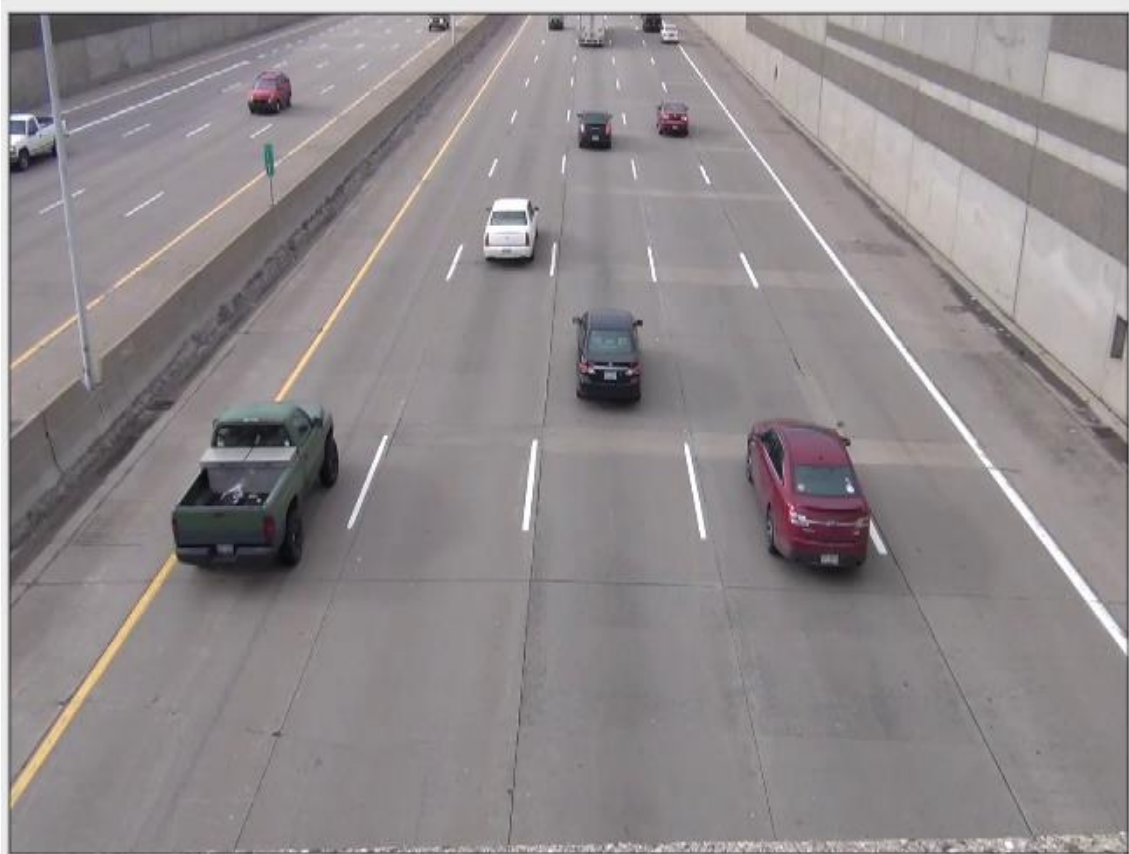


## IV. RESULTADOS Y DISCUSIÓN

### 4.1. RESULTADOS

Después de aplicar la metodología explicada en el capítulo anterior, pasamos a probar y ver los resultados de los algoritmos realizados.

En las figuras 18 y 19, se pueden observar las imágenes, previa (captura anterior) y actual (captura actual).



*FIGURA 18 CAPTURA DE IMAGEN PREVIA*

Se puede observar en la figura 19 los vehículos desplazados con respecto a la imagen de la figura 18. Como la cámara captura a 29 cuadros por segundo, se puede decir que el intervalo entre cuadro y cuadro es de 34.5 mSeg.



*FIGURA 19 CAPTURA DE IMAGEN ACTUAL*

Este desplazamiento depende de la velocidad con que vayan los vehículos. A continuación, se aprecia los cambios de color a escala de grises, como mencionamos, no interesa trabajar con color, ya que la información o los datos se reduce 3 veces. Si trabajamos con color serían  $1280 \times 740 \times 3 = 2'841,600$  pixeles por imagen y si trabajamos con escala de grises serían  $1280 \times 740 = 947,200$  pixeles por imagen. Posteriormente se aplica a cada imagen un filtro suavizante, pasa bajos, con la finalidad de eliminar ruido y difuminar las imágenes. Luego se restan estas imágenes, la previa y la actual, y el procesamiento continúa en base a la imagen resultado de esta resta. Los procesamiento posteriores son binarización, erosión, dilatación, etiquetado.



*FIGURA 20 IMAGEN PREVIA EN ESCALA DE GRISES*



*FIGURA 21 IMAGEN ACTUAL EN ESCALA DE GRISES*



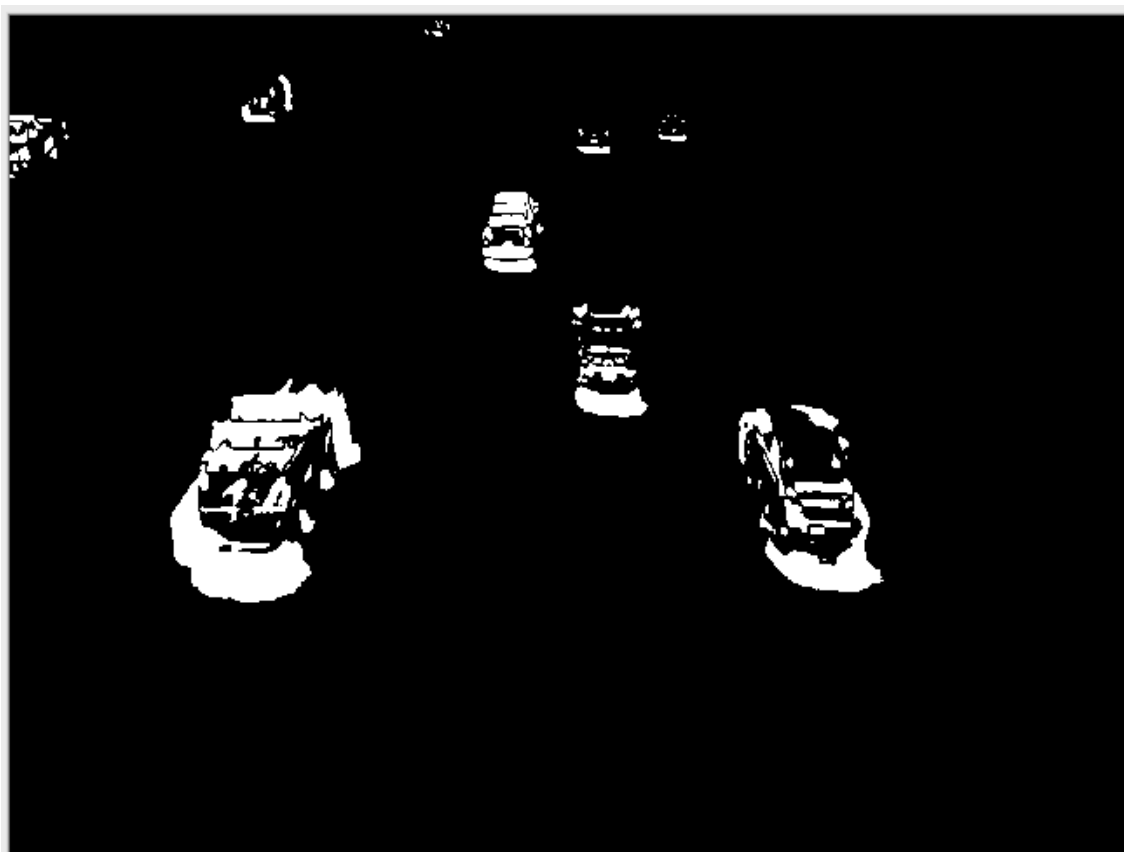
*FIGURA 22 IMAGEN PREVIA DESPUES DE APLICAR FILTRO*



*FIGURA 23 IMAGEN ACTUAL DESPUES DE APLICAR FILTRO*

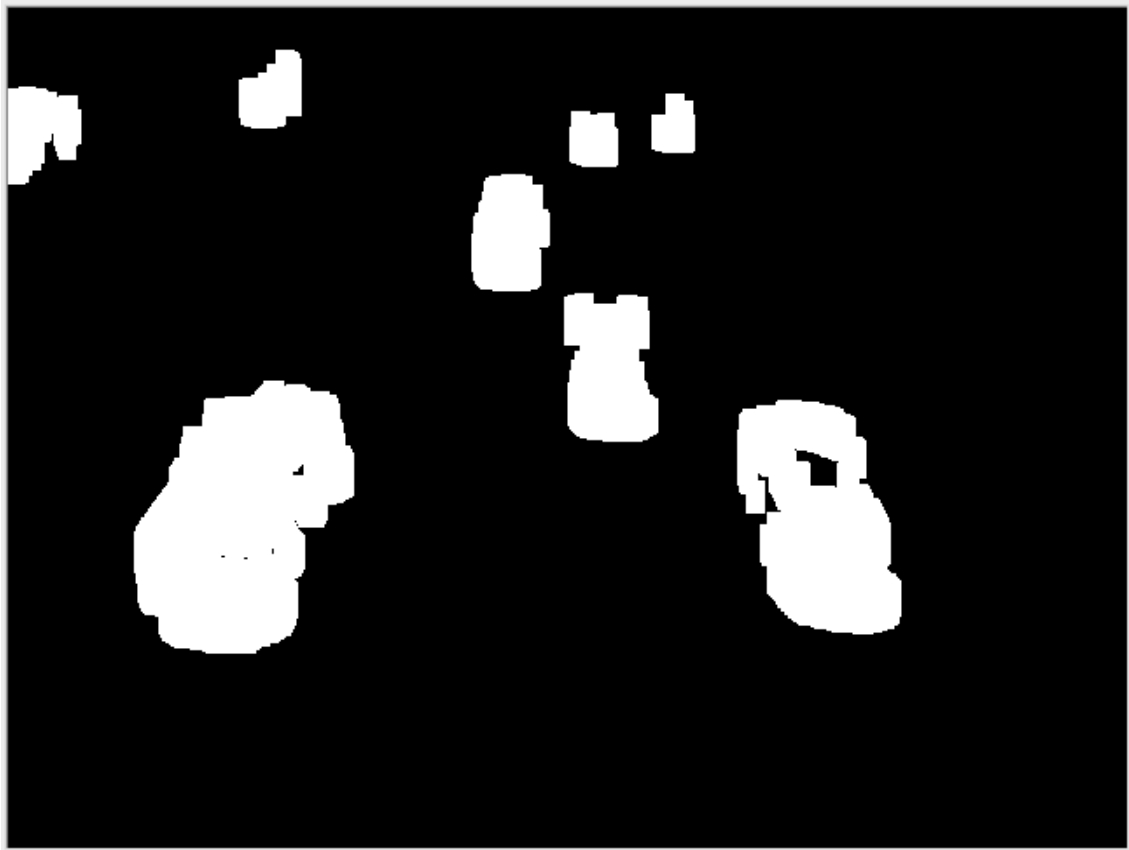


*FIGURA 24 RESULTADO DE LA DIFERENCIA ENTRE IMAGEN PREVIA Y ACTUAL*

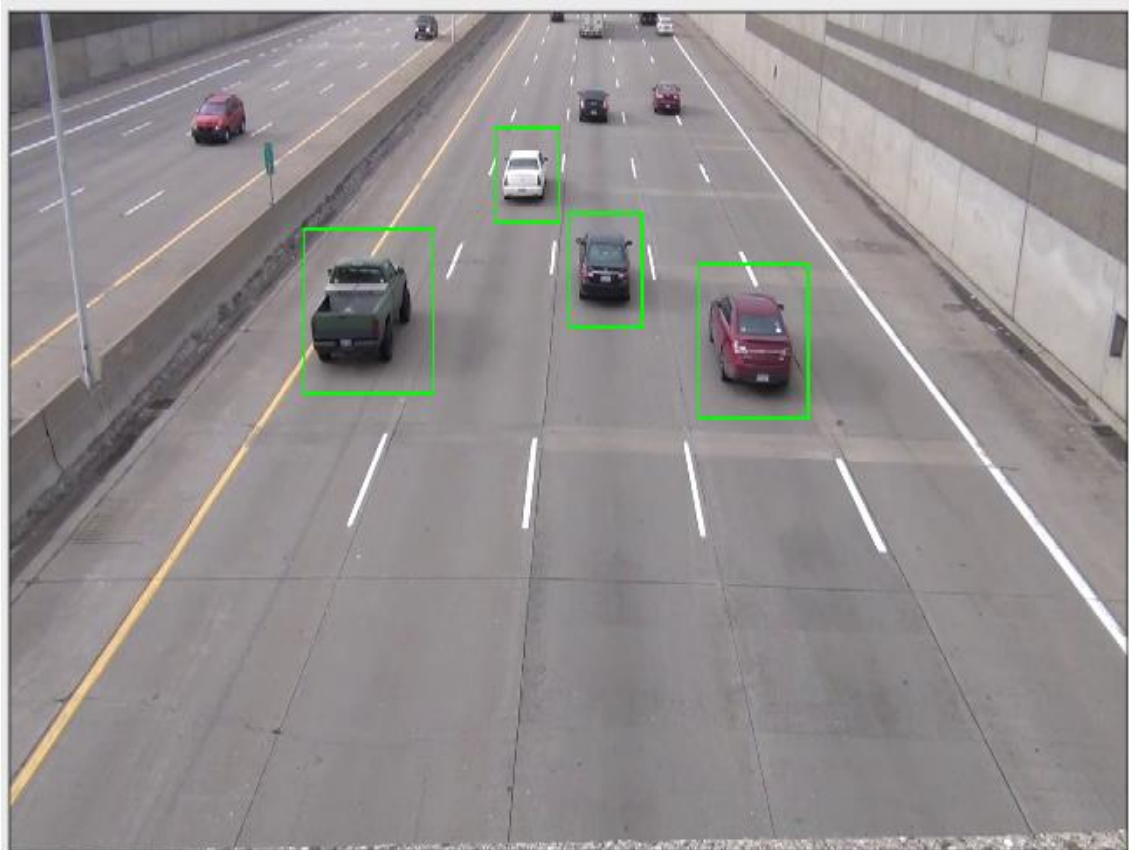


*FIGURA 25 IMAGEN BINARIZADA*

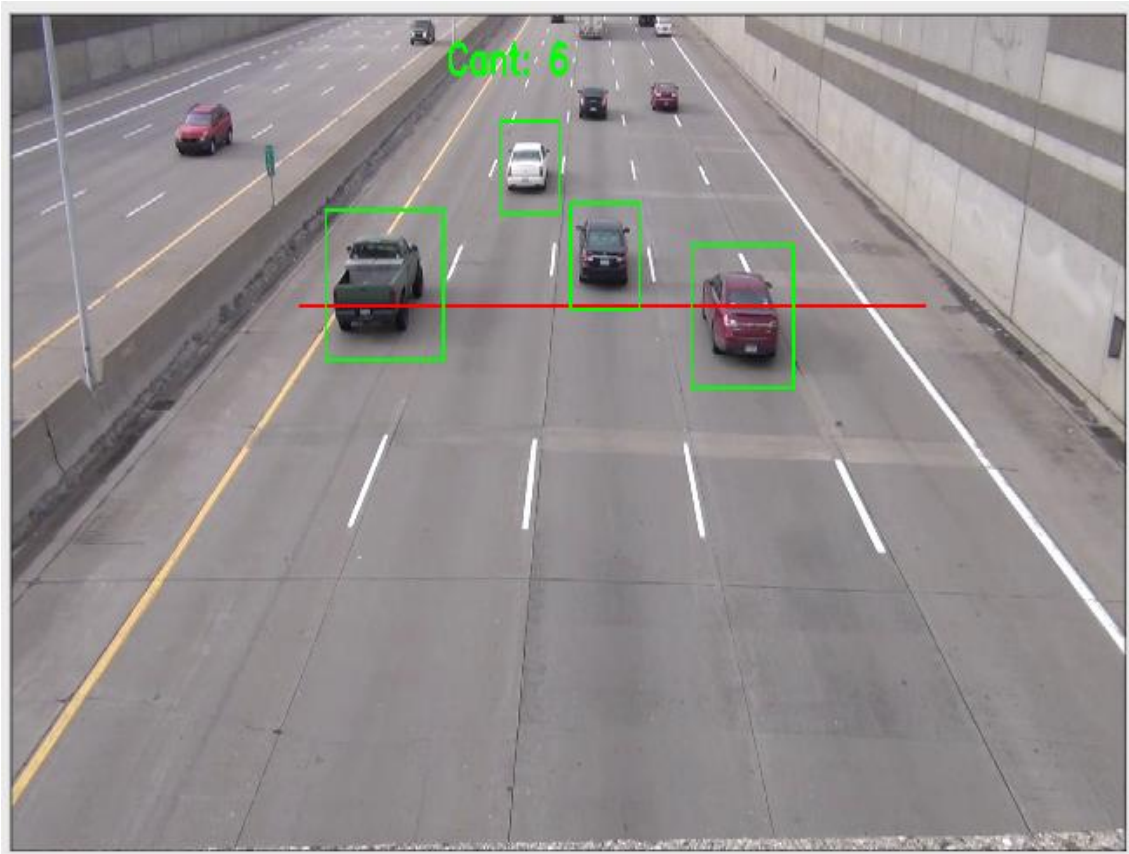




*FIGURA 26 IMAGEN DESPUES DE APLICAR EROSION Y DILATACIÓN*



*FIGURA 27 IMAGEN CON REGIÓN DE VEHÍCULOS DETECTADOS*



*FIGURA 28 IMAGEN CON REGIÓN DE VEHÍCULOS: DETECCIÓN Y CONTEO*

En esta última figura 28, se puede ver el resultado final, la detección y conteo de vehículos. Cabe precisar, que para que los algoritmos funcionen se necesita que la cámara se encuentre fija, de lo contrario, todo será captado como movimiento.

## **4.2. DISCUSIÓN**

La metodología empleada es una alternativa a muchas existentes, ya que se puede observar en las imágenes, el resultado desde la adquisición en el modelo RGB, conversión a escala de grises, filtrado pasa bajo, diferencia de imágenes, la binarización, erosión y dilatación, y etiquetado que nos permite asignarle un nombre a cada objeto y en base a esto, saber el área y la posición de cada objeto. Para la metodología empleada, mucho tiene que ver con los resultados la iluminación, no es lo mismo de día que de noche. La parte de binarización, podemos ver, que necesita un umbral, este valor cambia de acuerdo a la iluminación.



## CONCLUSIONES

El propósito del proyecto de tesis se ha logrado, se implementó un sistema de detección y conteo de vehículos utilizando visión artificial.

El presente trabajo plantea un algoritmo que permite realizar conteo vehicular en tiempo real, para su procesamiento se utiliza un computador con prestaciones comunes y se obtiene un alto grado de desempeño.

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código C++ optimizados, aprovechando además las capacidades que proveen los procesadores multi-núcleo. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel.

Un gran número de empresas a nivel mundial se ha dedicado a proveer soluciones integrales a problemas de la industria utilizando como herramienta fundamental la visión artificial, esto la ha convertido en un área de gran desarrollo. Se han implementado ampliamente sistemas enfocados al estudio y caracterización del tráfico.

## **RECOMENDACIONES**

Para implementar el sistema existen diferentes métodos, que tendrían que probarse, para la mejora de la eficiencia de los algoritmos en cuanto a tiempo de procesamiento. Por ejemplo, una opción puede ser, procesar pequeñas regiones solamente por donde pasan los vehículos. Esto aumentaría la velocidad de procesamiento y habría menos información que procesar.

Aplicaciones se le pueden dar muchas a este sistema, por ejemplo, análisis de tráfico en las calles, para tomar decisiones en los semáforos. Esto le puede servir a las Municipalidades.

De acuerdo con la aplicación se hará un sistema posterior a éste.

## REFERENCIAS BIBLIOGRÁFICAS

1. MERY, Domingo. Visión por Computador. Departamento de ciencia de la Computación Universidad Católica de Chile. Santiago de Chile, 2004
2. Rafael C. Gonzales, Richard E. Woods, Tratamiento Digital de Imágenes. USA, Addison Wesley Iberoamericana, S. A. 1996. 755 pp.
3. Michael Alder, An Introduccion to Pattern Recognition, Mike Alder, 2001.
4. Abraham Kandel, Horst Bunke, Mark Last, Applied graph theory in computer visión and pattern recognition, Springer-Verlag Berlin Heidelberg, 2007.
5. Bernd Jahne, Digital image processing, Springer-Verlag Berlin Heidelberg, 2002.
6. Gonzalo Pajares Martinsanz, Jesús Manuel de la Cruz García, José Manuel molina pascual, Juan Cuadrado Pardo, Alejandro López Correa, “Imágenes digitales - Procesamiento practico con java, Alfa Omega Grupo Editor S.A. 2004
7. Computer vision with the OpenCV library; Gary Bradsky y Adrian Kaebler.
8. Manual original proporcionado por intel: Intel® Open Source Computer Vision Library.
9. Learning OpenCV: Computer Vision with the OpenCV Library, Bradski, G.,
10. BRADSKY, Gary, KAEHLER, Adrián. Learning OpenCv.
11. GONZÁLEZ, Juan Manuel et al. ¿Cómo mejorar el flujo vehicular por medio de la simulación? Saltillo Coahuila, México Escuela Nacional de Optimización y Análisis Numérico, 2003.

## ANEXOS

### ANEXO 1: RESULTADO DE MAS CAPTURAS DEL SISTEMA

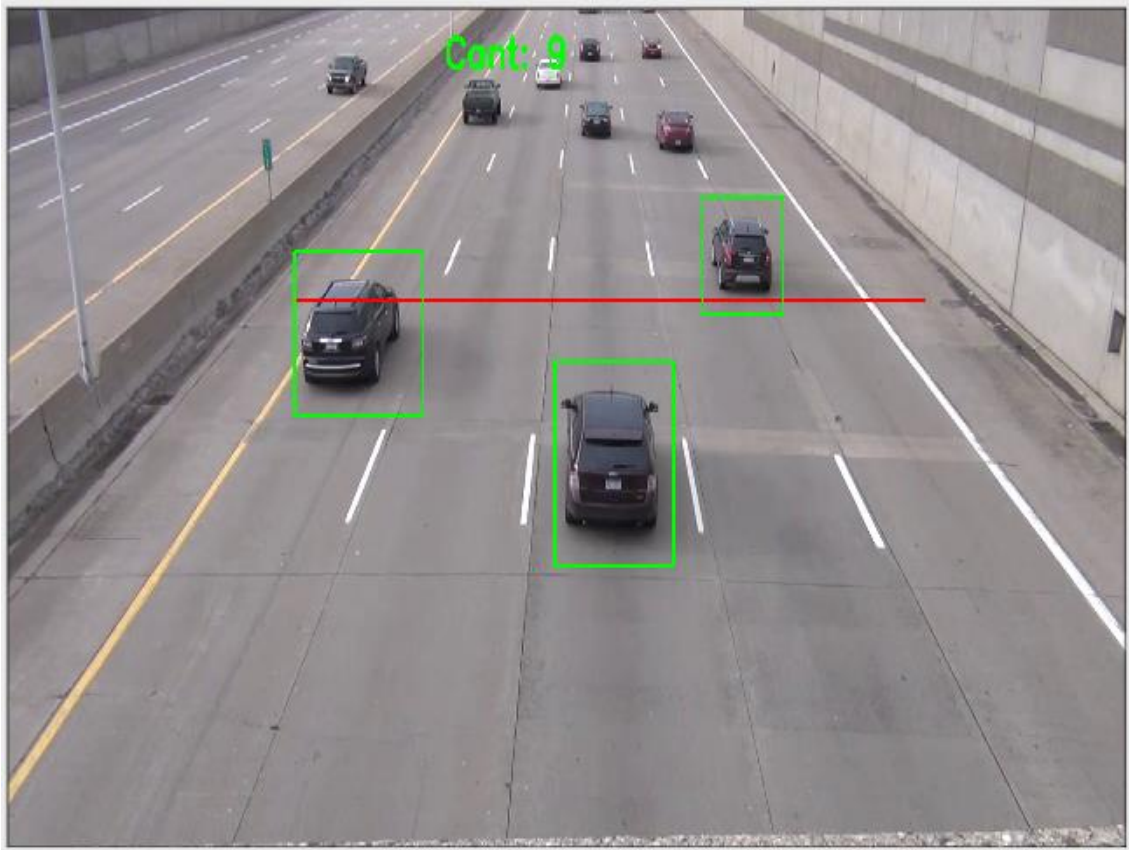
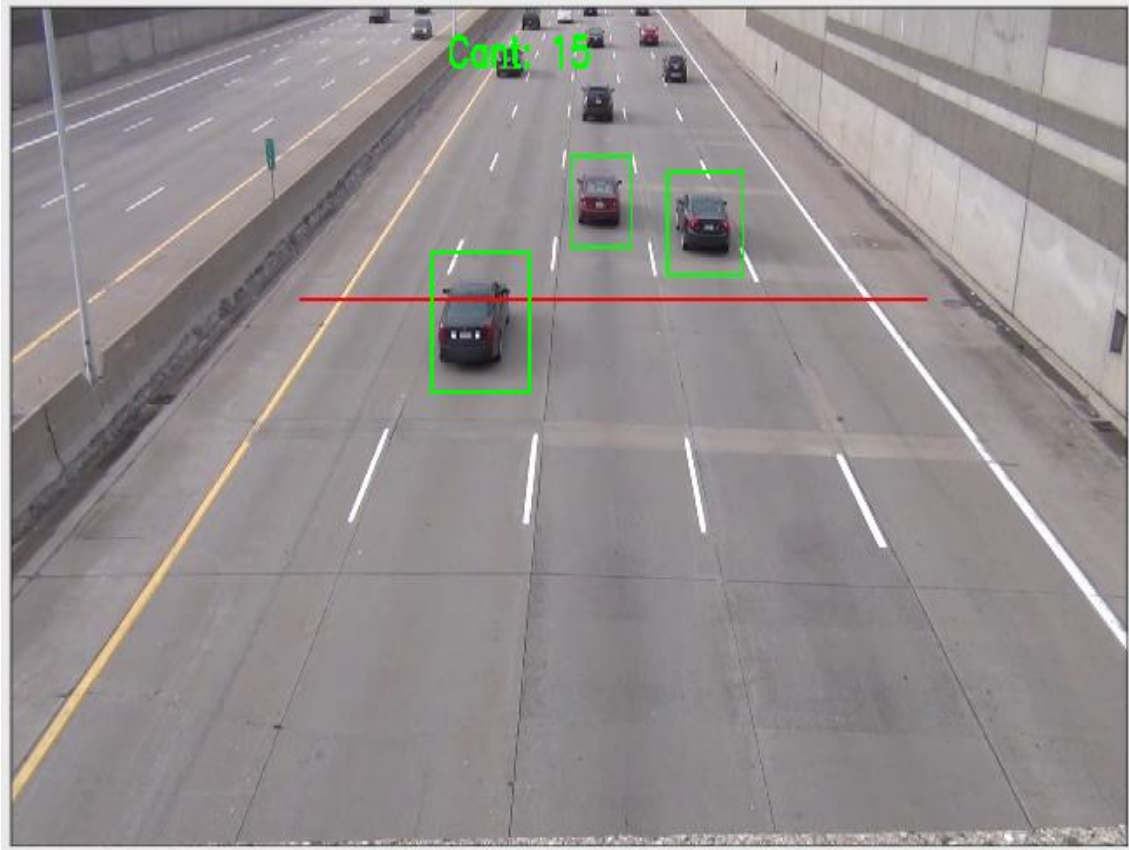
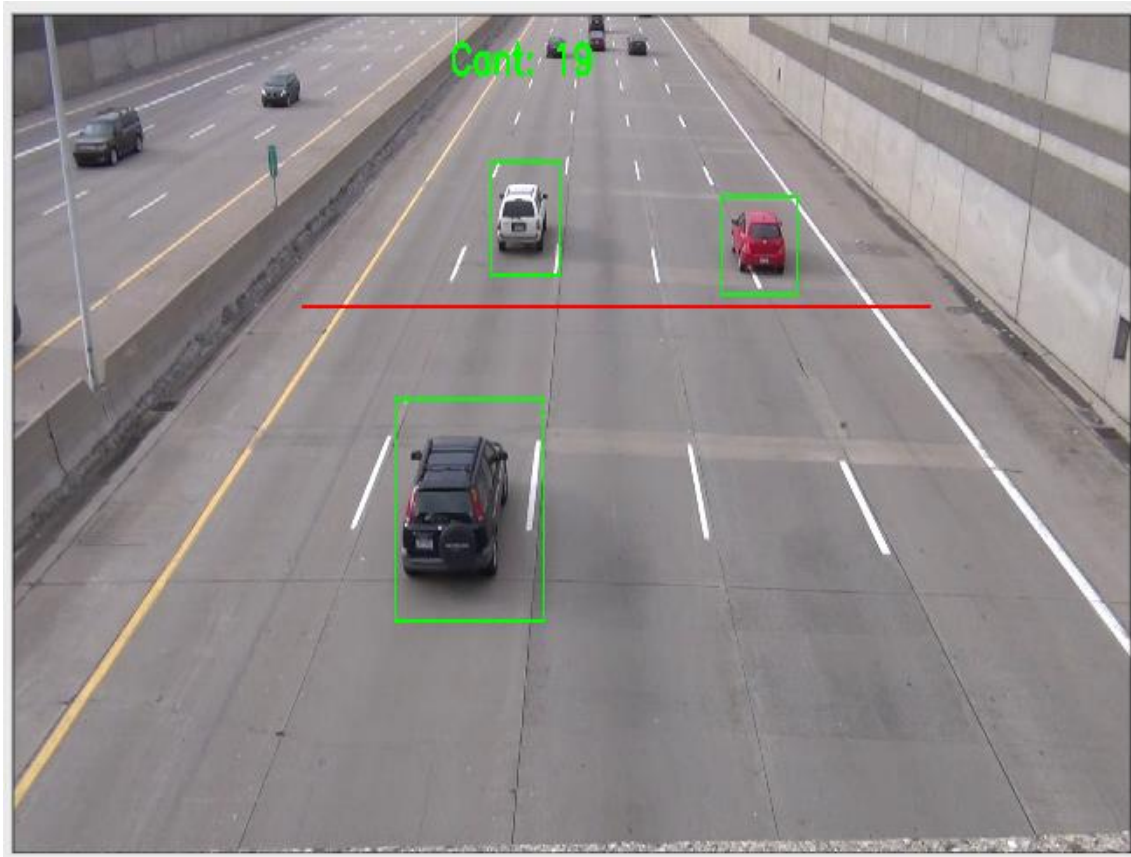


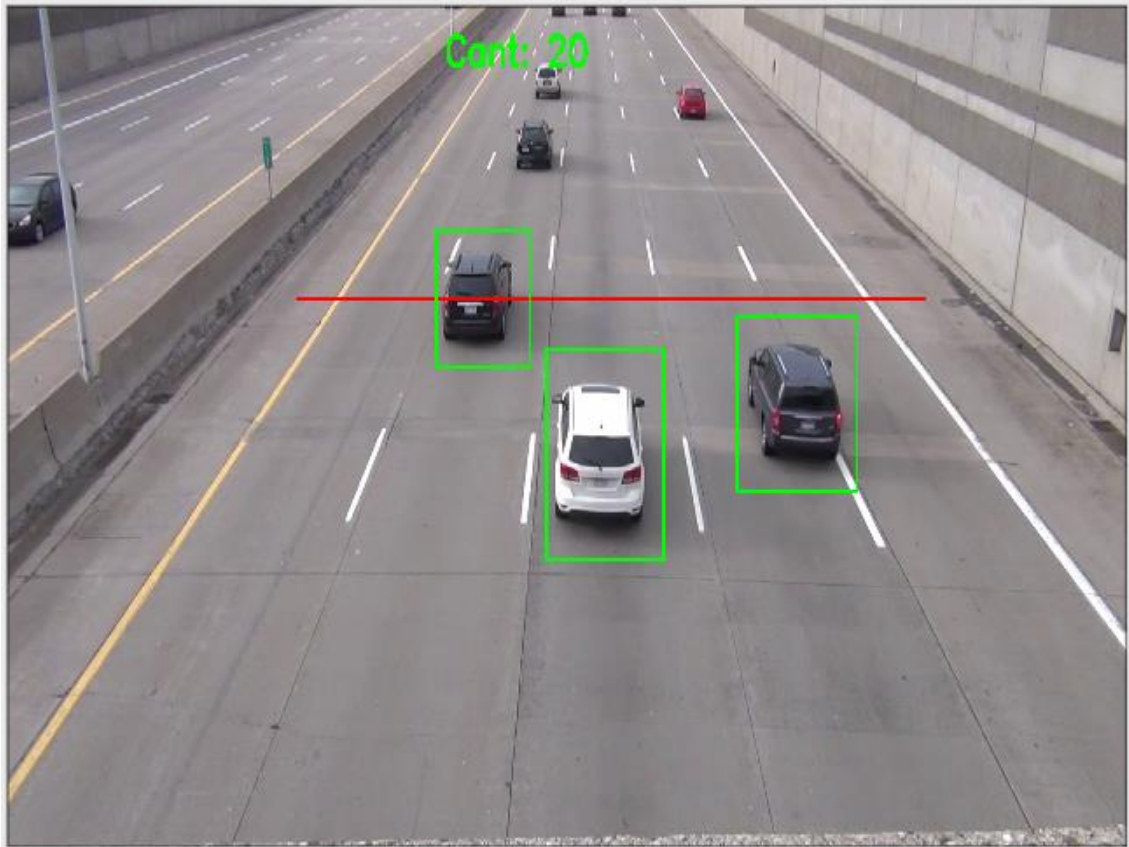
FIGURA 29: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS



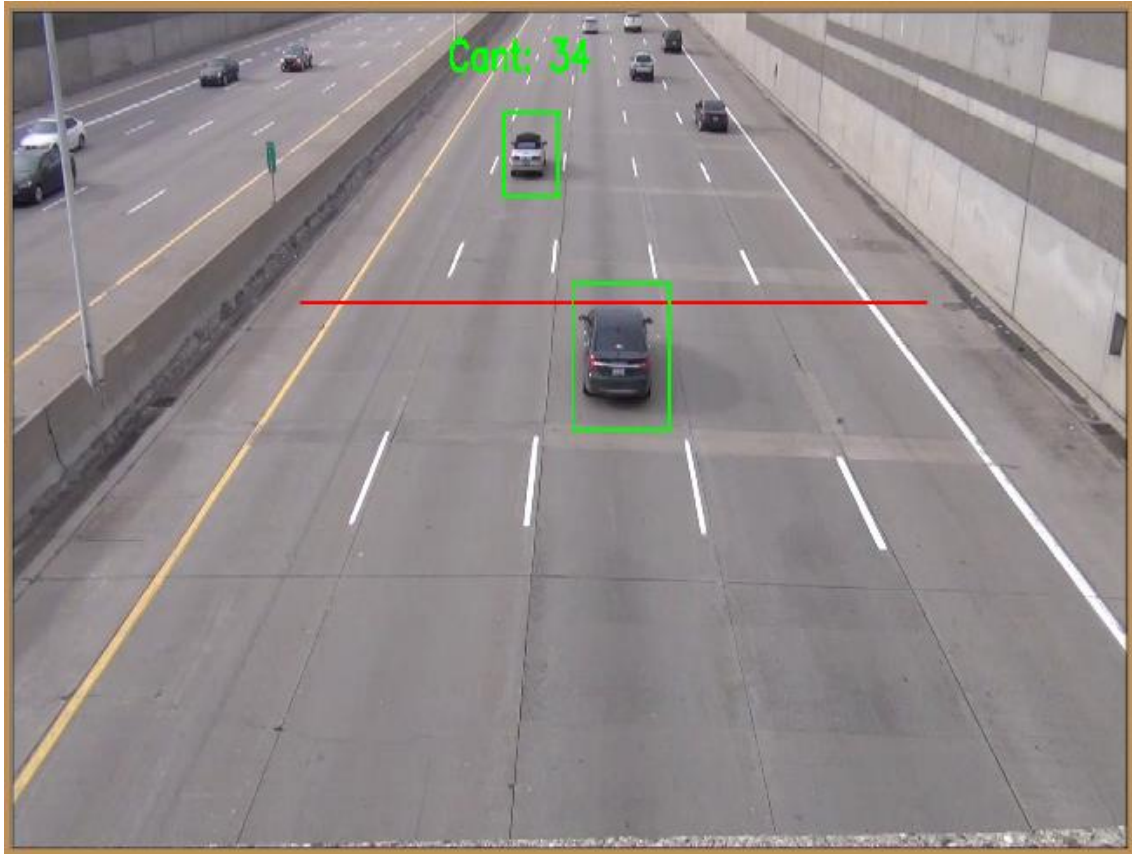
*FIGURA 30: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS*



*FIGURA 31: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS*

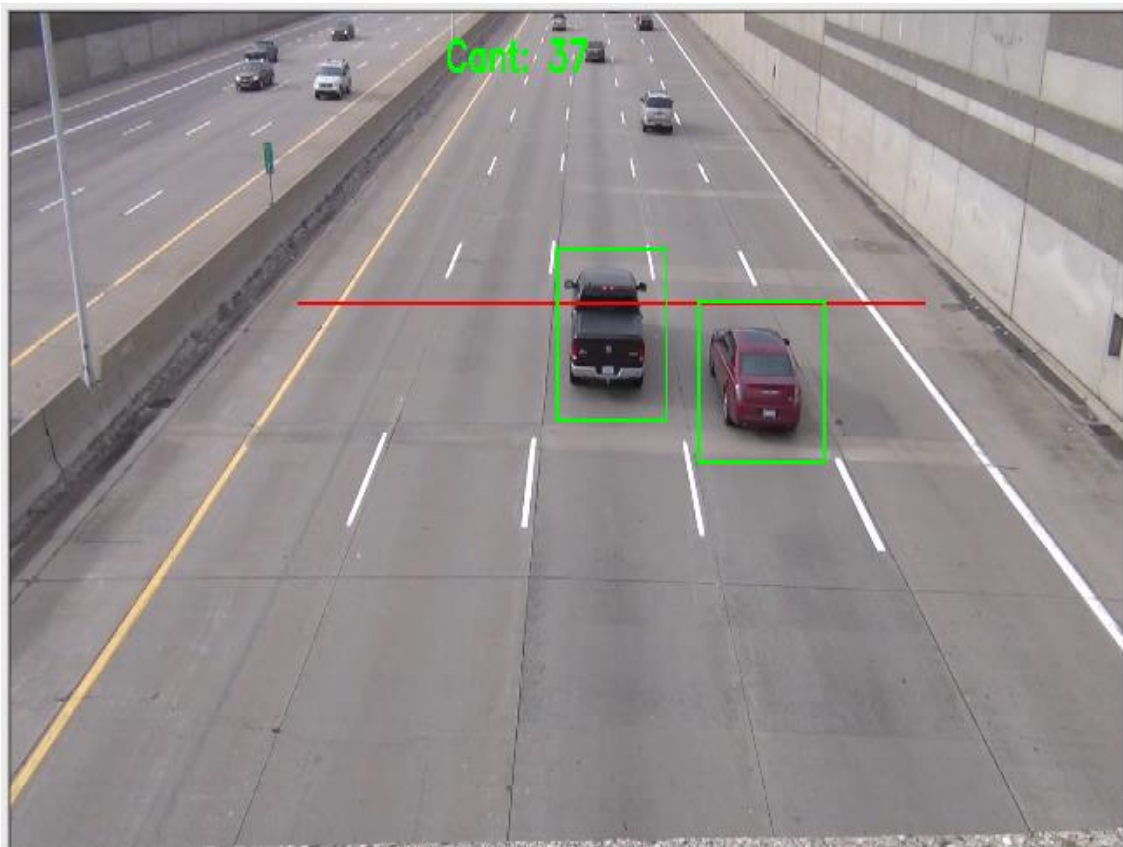


*FIGURA 32: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS*

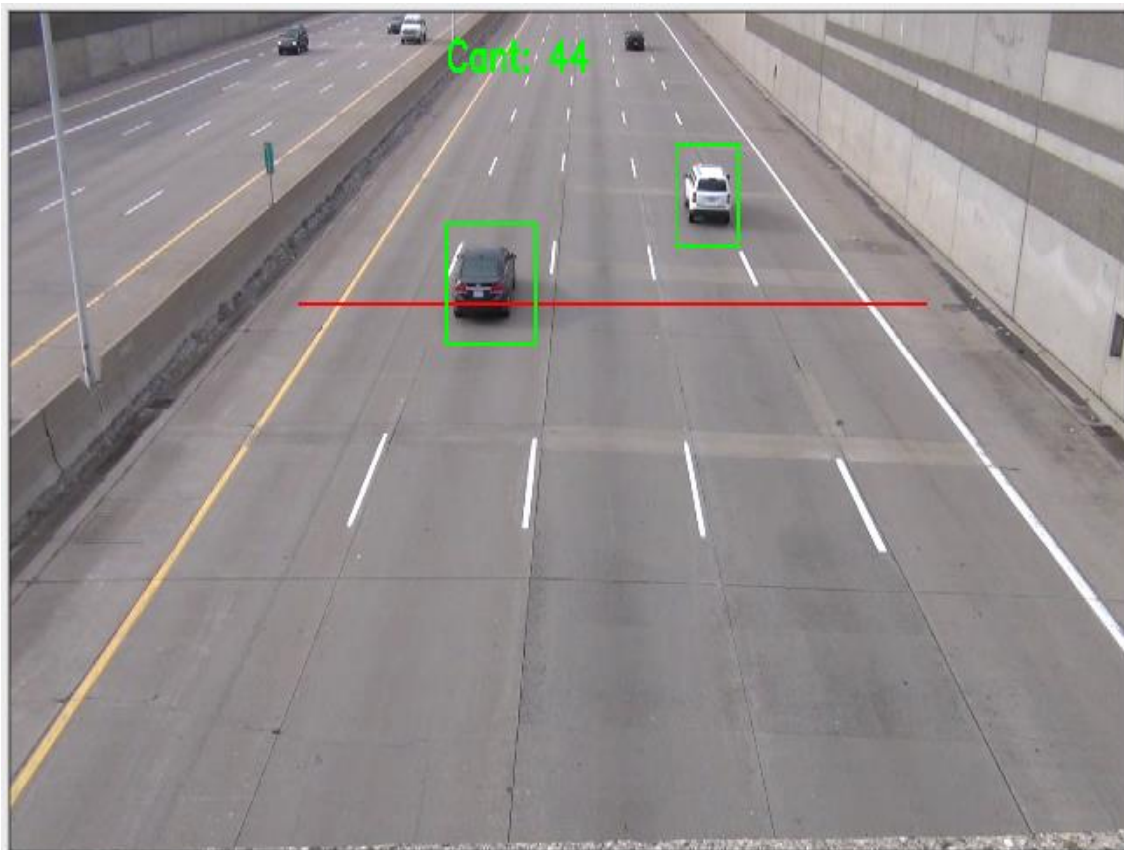


*FIGURA 33: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS*





*FIGURA 34: RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS*



*FIGURA 35 RESULTADO DE LA DETECCIÓN Y CONTEO DE VEHÍCULOS*

En la instalación se utiliza OpenCV 2.4.5 y Visual Studio 2012. Pero es casi lo mismo para otras versiones de OpenCV y Visual Studio.

Hay 2 maneras de instalar OpenCV en la computadora. La forma más adecuada es la instalación mediante el uso de las bibliotecas preconstruidas. Por lo tanto, voy a discutir cómo instalar OpenCV mediante bibliotecas preconstruidas.

En primer lugar debe tener un IDE adecuado, instalado Visual Studio Express 2012 que es una edición gratuita. (Registrarse dando la dirección de correo electrónico para obtener la clave de producto gratuita).

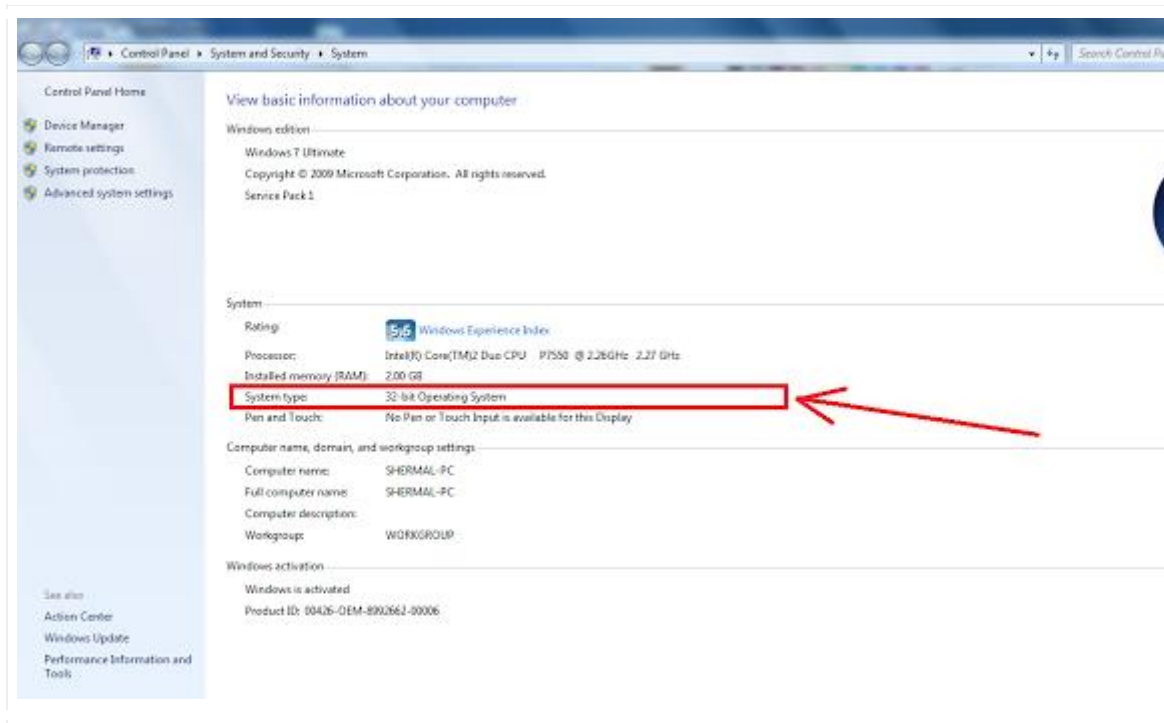
Descargar OpenCV y elegir la versión OpenCV 2.4.5.

A continuación, haga doble clic en el archivo descargado "OpenCV-2.4.5.exe".

Ahora configurar correctamente las variables de entorno para poder usar OpenCV.

### Pasos para configurar la variable de entorno

- Hacer clic derecho en '**Mi PC**' y luego en '**Propiedades**' en el menú desplegable. Se verá una ventana como esta.



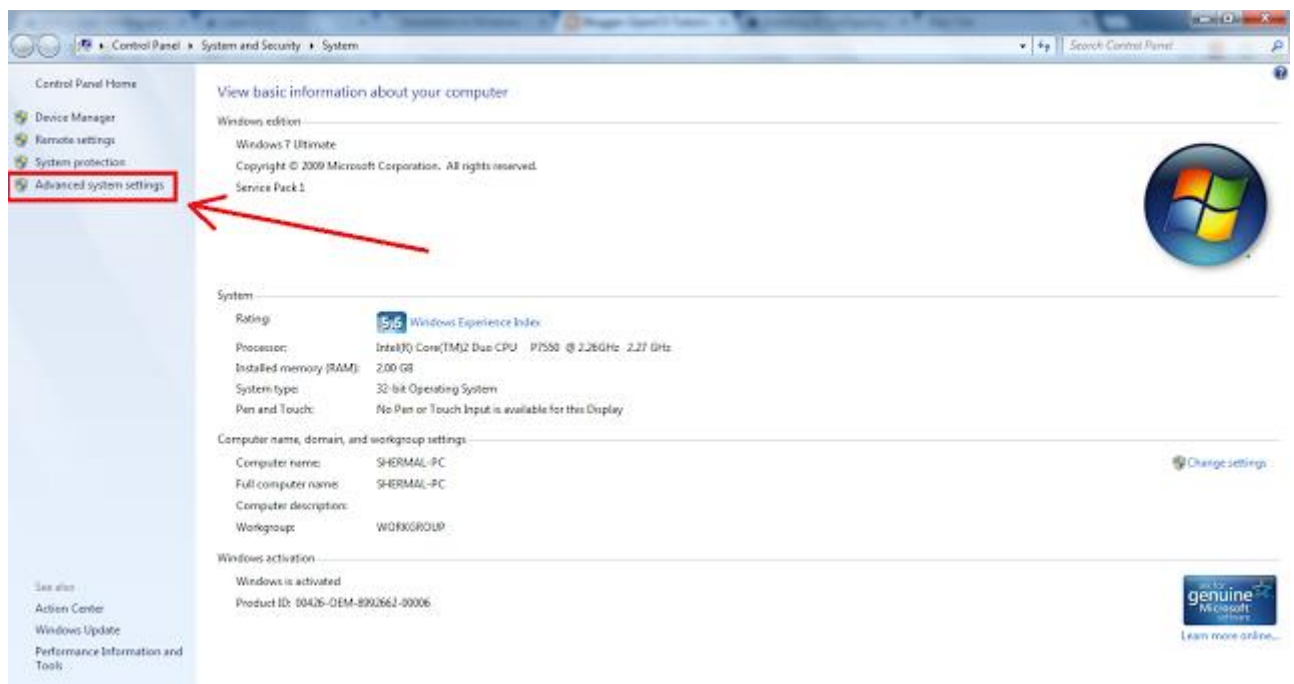
Aquí el tipo de sistema es el sistema operativo de **32 bits**. Así que la arquitectura del

sistema es **x86**. Si el tipo de sistema es el sistema operativo de **64 bits**, la arquitectura del sistema es **x64**.

Tanto Visual Studio 2012 como OpenCV2.4.5 admiten vc11. Por lo tanto, en este tutorial, voy a utilizar vc11 como el tipo de compilador. Puede encontrar los tipos de compilador compatibles de OpenCV, si va a C: \ opencv \ build \ x86 o C: \ opencv \ build \ x64 de acuerdo con la arquitectura de su sistema.

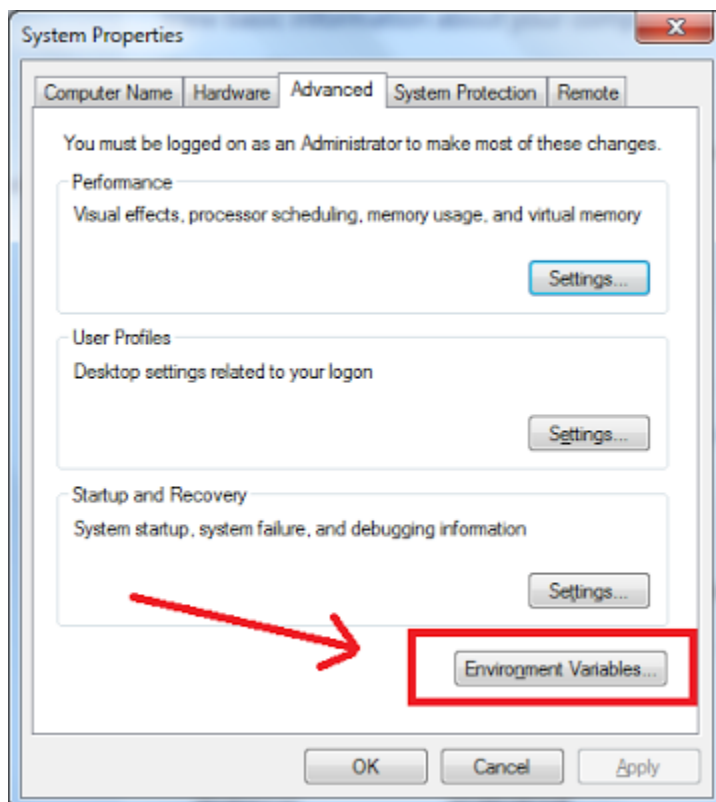
Esto se puede hacer en la línea de comandos o mediante GUI.

Hacer clic derecho en ' **Mi PC** ' y luego en ' **Propiedades** ' en el menú desplegable. Se verá una ventana como esta.

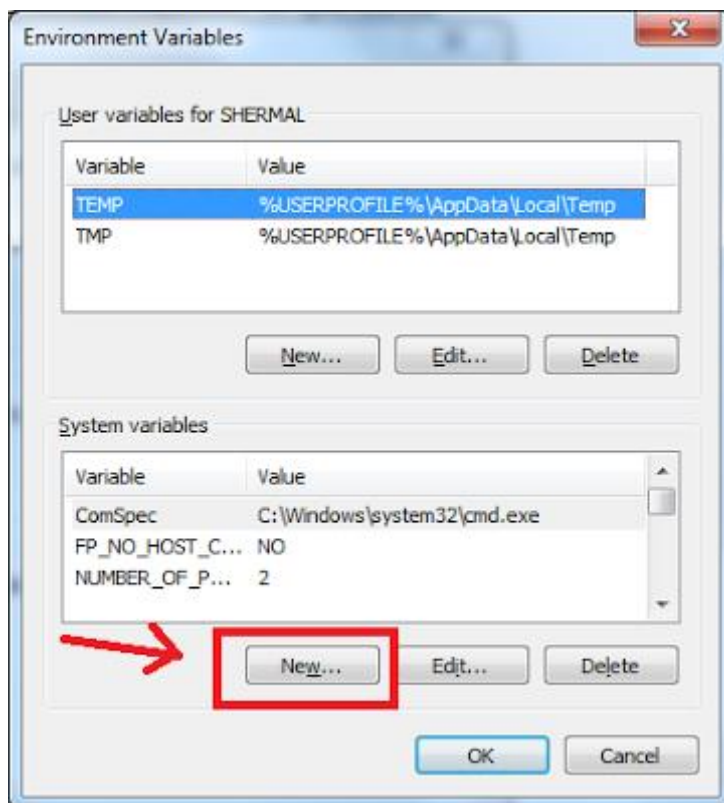


Haga clic en " **Avanzar configuración del sistema**" en la ventana anterior

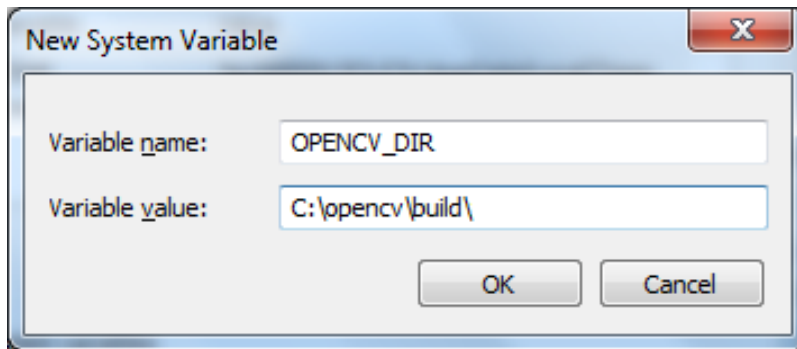
A continuación, hacer clic en "**Variables de entorno**"



A continuación, haga clic en el botón "Nuevo" en la parte inferior de la ventana



Tipo **OPENCV\_DIR** contra el **Nombre de la variable**. Y escribir la ubicación **C: \opencv \ build \** contra el **valor Variable**.

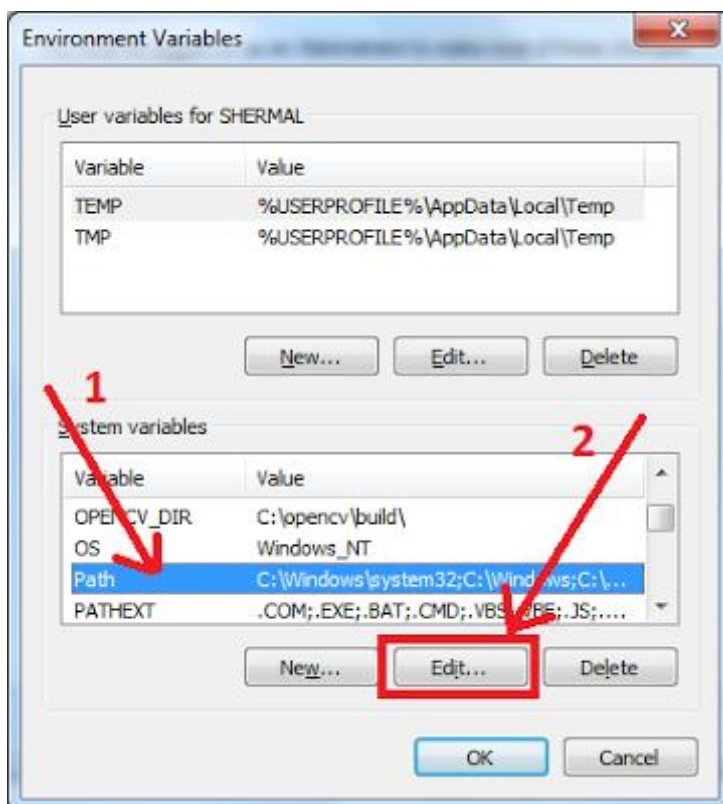


A continuación, pulsar **OK**

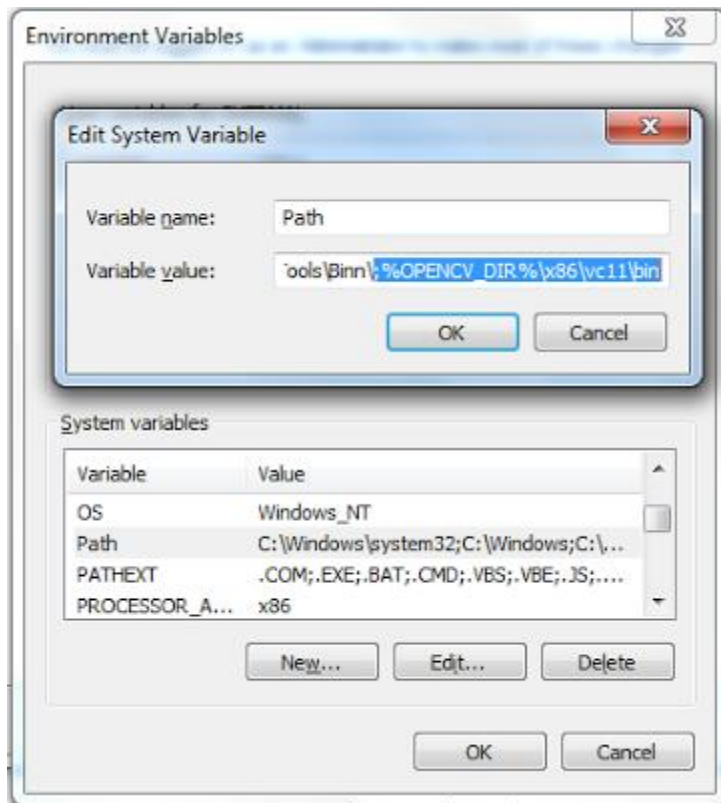
Ahora ha agregado una nueva variable de entorno al sistema.

Ahora se edita una variable del sistema.

Haga clic en '**PATH**' dentro de la lista de variables del sistema y luego hacer clic en el botón '**Editar**' en la parte inferior de la ventana



A continuación, agregar un; al final de la línea agregar lo siguiente % OPENCV\_DIR% \ x86 \ vc11 \ bin, después del punto y coma. X86 es la arquitectura del sistema y vc11 es el tipo de compilador.

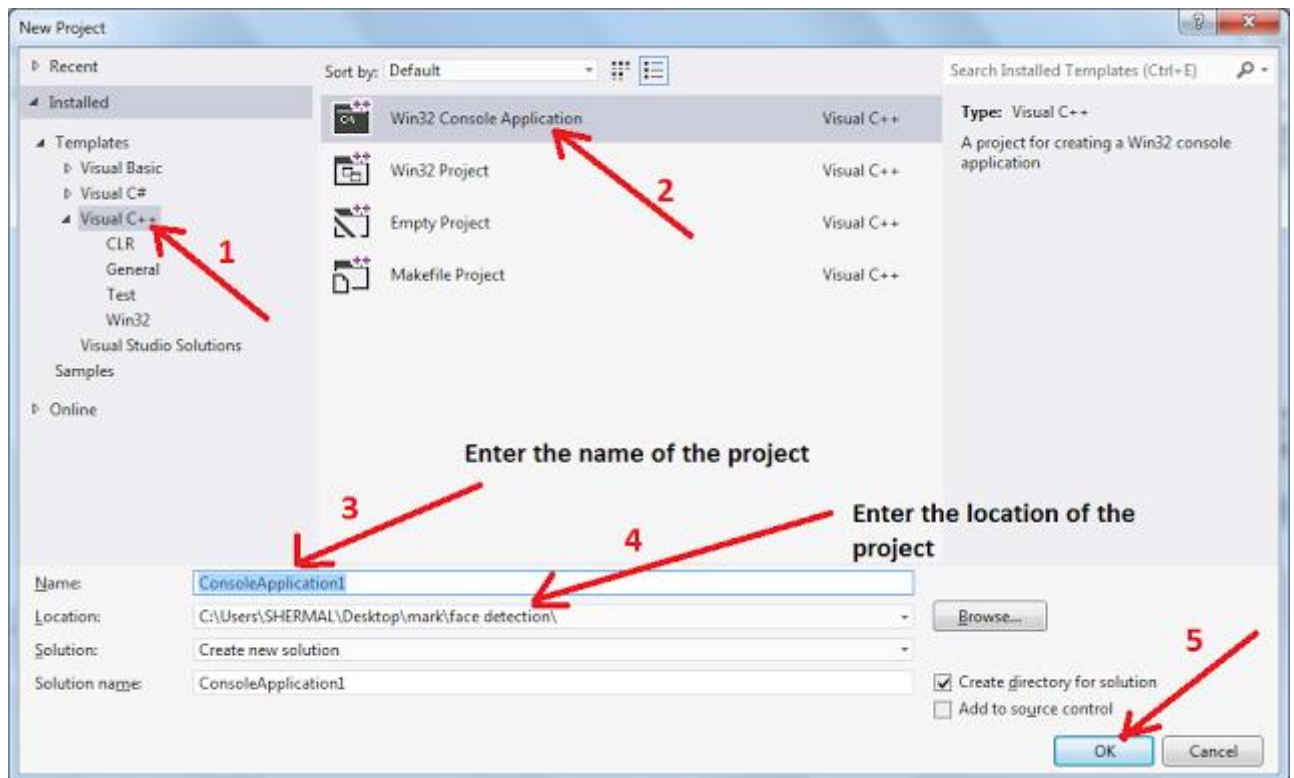


Ahora hay que configurar Visual Studio.

### Configurar Visual Studio Express

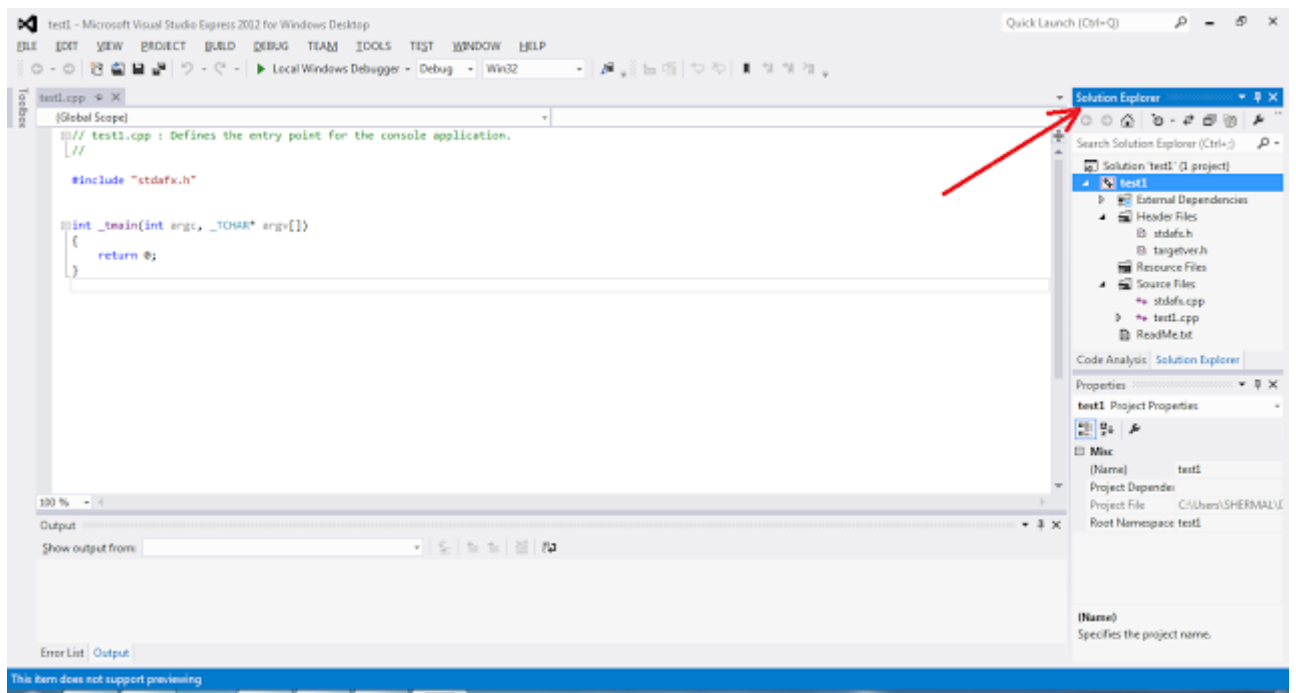
- Iniciar Microsoft Visual Studio
- Ir a Archivo> Nuevo proyecto ...
- Hacer todo lo que se muestra en la imagen de abajo y hacer clic en Aceptar y, a continuación, hacer clic en Finalizar.



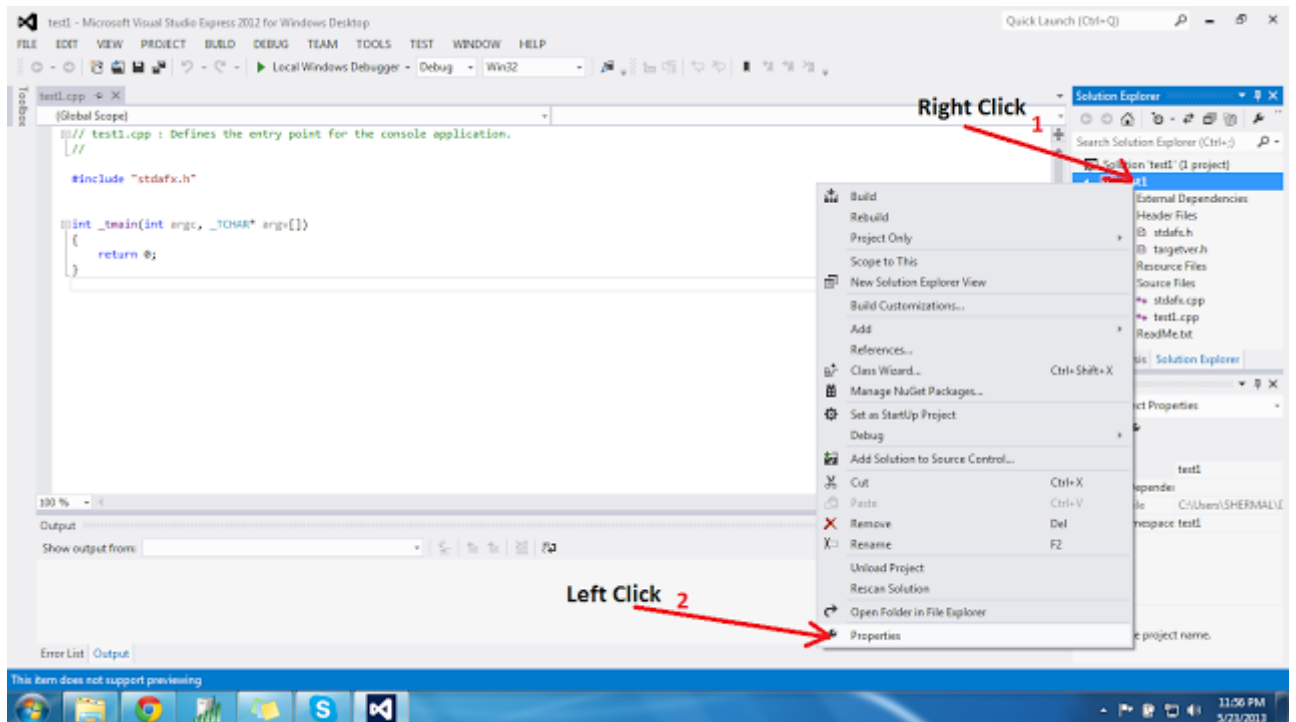


- Presionar la **tecla 'Ctrl'** y luego **'W'** mientras mantiene presionada la tecla **'Ctrl'** en su teclado. Soltar ambas teclas. A continuación, pulsar **'S'** en el teclado. A continuación, se verá el panel **"Explorador de soluciones"** en el lado derecho o izquierdo de la ventana de Visual Studio.

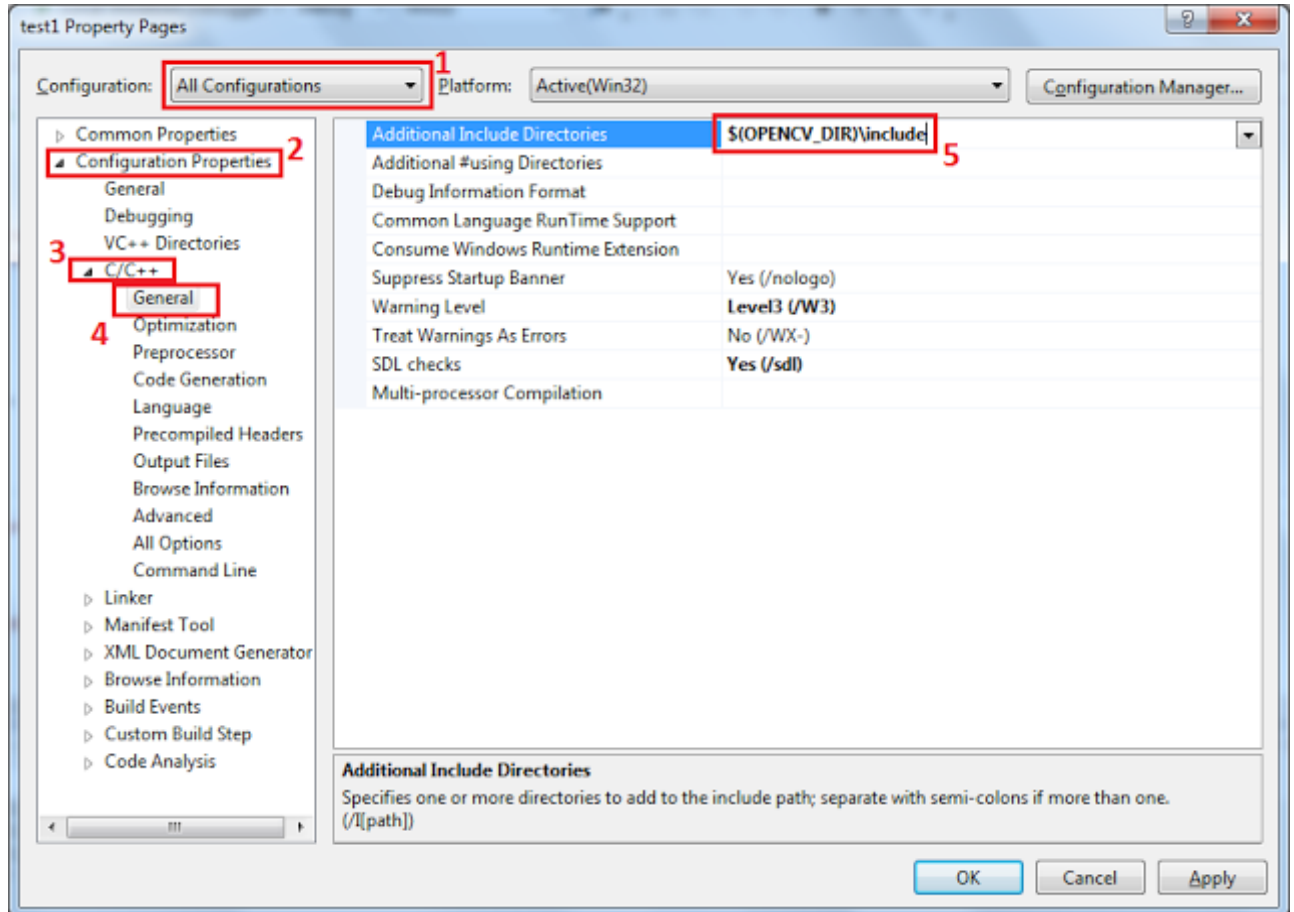




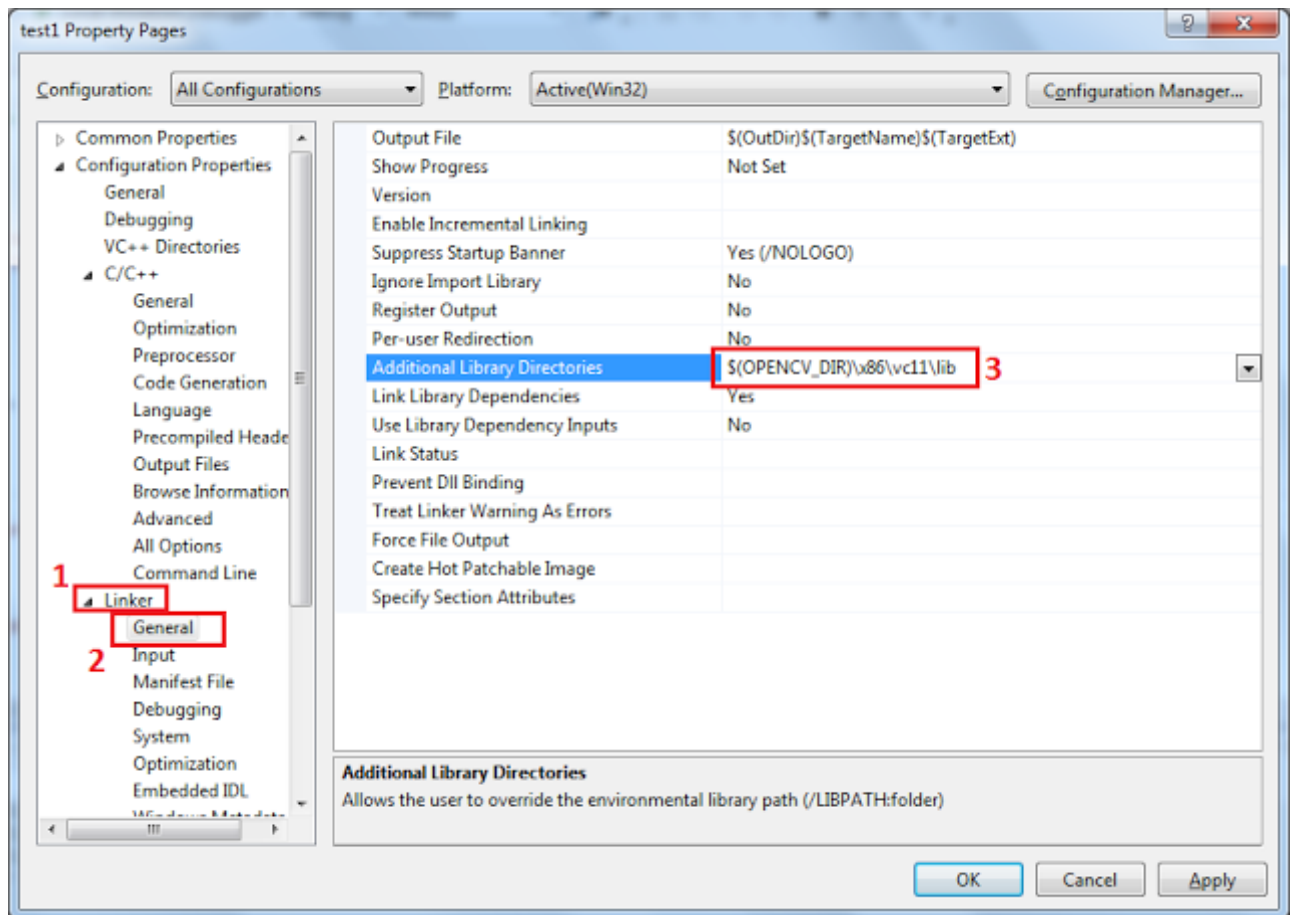
- A continuación, hacer clic con el botón secundario en el nombre del proyecto (ha introducido este nombre en un paso anterior) y, a continuación, hacer clic en **"Propiedades"**



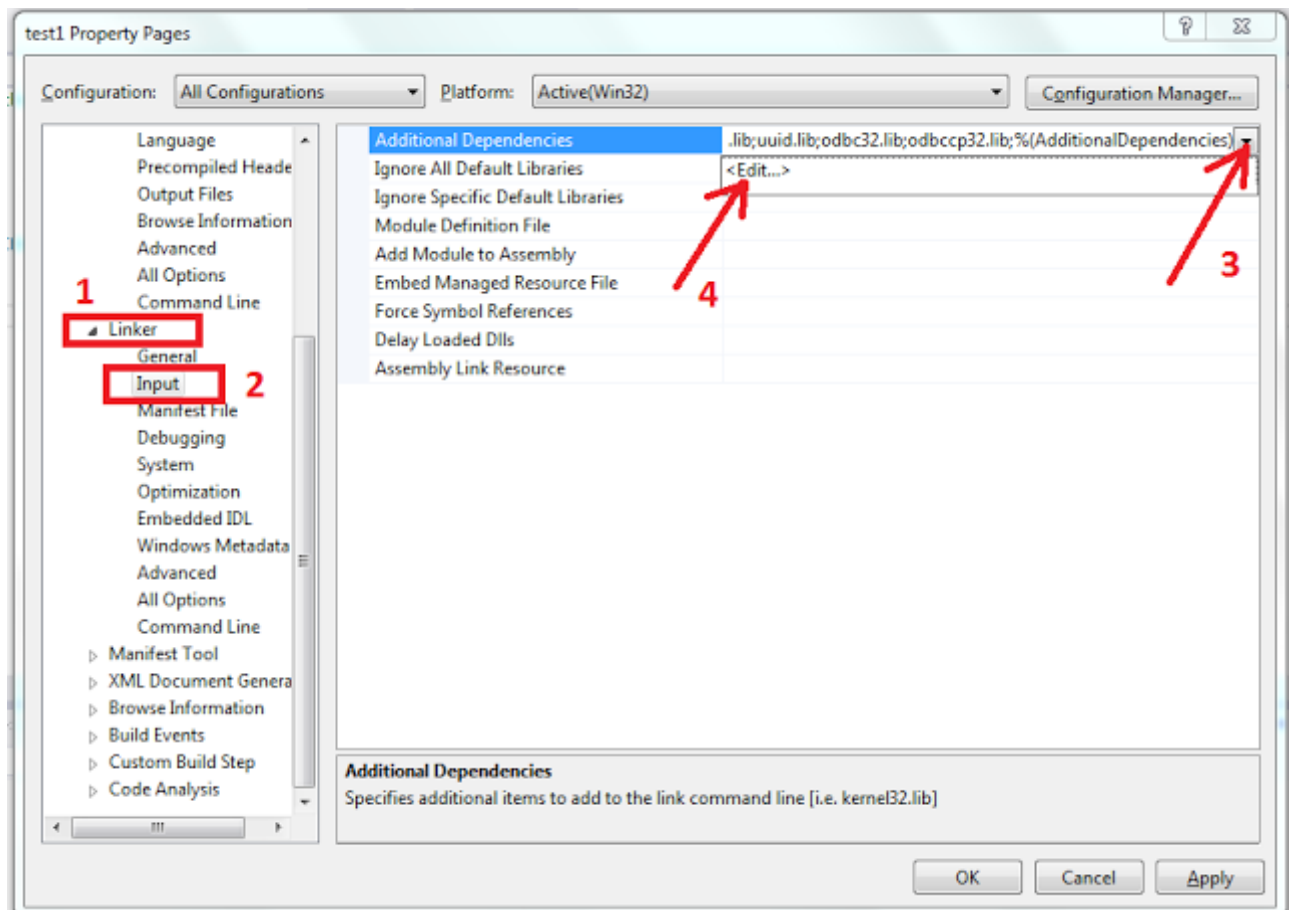
- Seleccionar 'Todas las configuraciones' como se indica en la 1ª casilla. Hacer clic en los 3 lugares indicados por 3 cuadros siguientes en el orden dado. A continuación, copiar y pegar \$ (OPENCV\_DIR) \ include en la opción de 'Incluir Directorios Adicionales'



- Hacer clic en los primeros 2 lugares como se indica por los primeros 2 cuadros en el orden dado. Luego, copiar y pegar \$(OPENCV\_DIR) \ x86 \ vc11 \ lib en la opción "Directorios de biblioteca adicionales"



Hacer clic como se muestra a continuación



Después de hacer clic en <Editar ...> (4ª flecha en la imagen anterior), aparecerá un cuadro de diálogo y se tendrá que copiar y pegar los siguientes nombres de archivo de biblioteca. Si se está utilizando una versión diferente de OpenCV que no sea 2.4.5, comprobar la ubicación 'C: \ opencv \ build \ x86 \ vc11 \ lib' para los nombres de archivo respectivos.

Opencv\_calib3d245d.lib

Opencv\_contrib245d.lib

Opencv\_core245d.lib

Opencv\_features2d245d.lib

Opencv\_flann245d.lib

Opencv\_gpu245d.lib

Opencv\_haartraining\_engined.lib

Opencv\_highgui245d.lib

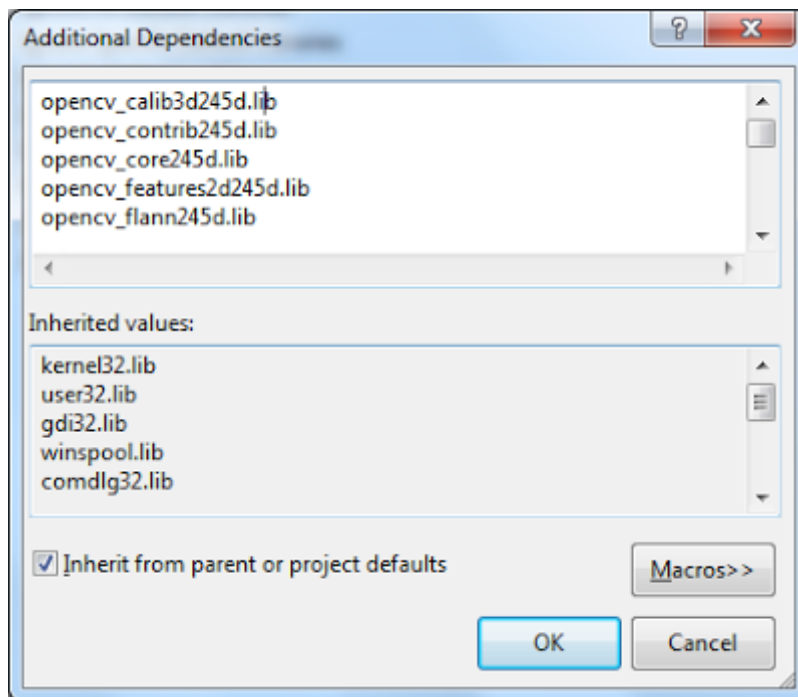
Opencv\_imgproc245d.lib

Opencv\_legacy245d.lib

Opencv\_ml245d.lib

Opencv\_nonfree245d.lib  
Opencv\_objdetect245d.lib  
Opencv\_photo245d.lib  
Opencv\_stitching245d.lib  
Opencv\_superres245d.lib  
Opencv\_ts245d.lib  
Opencv\_video245d.lib  
Opencv\_videostab245d.lib

Estos son algunos de los nombres de archivo, que se puede encontrar en la ubicación 'C:  
\opencv\build\x86\vc11'.



Ahora hacer clic en Aceptar

## PROGRAMA PRINCIPAL

```
#include <opencv2\opencv.hpp>
using namespace cv;
using namespace std;

int main()
{
    Mat frame1, frame2, imgDif, imgBin;
    Mat imgf1, imgf2;
    int car = 0, lim_veh = 0;
    //VideoCapture cap("vehiculos2.wmv");
    VideoCapture cap("vehiculos1.mp4");
    if(!cap.isOpened())
        return -1;
    namedWindow("Video",0);
    namedWindow("Salida",0);
    bool bSuccess = cap.read(frame1); // read a new frame from video
    if (!bSuccess) return 0;

    while(1)
    {
        bool bSuccess1 = cap.read(frame2); // read a new frame from video
        if (!bSuccess1) break;

        cvtColor(frame1, imgf1, CV_BGR2GRAY);
        cvtColor(frame2, imgf2, CV_BGR2GRAY);
        blur(imgf1, imgf1, cv::Size(5, 5));
        blur(imgf2, imgf2, cv::Size(5, 5));
        cv::absdiff(imgf1, imgf2, imgDif);
        threshold(imgDif, imgBin, 30, 255.0, CV_THRESH_BINARY);
    }
}
```

```

        morphologyEx(imgBin,          imgBin,          MORPH_ERODE,
getStructuringElement(MORPH_RECT,Size(5,1)),Point(-1,-1),1);
        morphologyEx(imgBin,          imgBin,          MORPH_DILATE,
getStructuringElement(MORPH_RECT,Size(5,7)),Point(-1,-1),4);
        morphologyEx(imgBin,          imgBin,          MORPH_DILATE,
getStructuringElement(MORPH_RECT,Size(3,3)),Point(-1,-1),2);

```

```

resizeWindow("Salida",640,480);
imshow("Salida", imgBin);

```

```

vector<vector<Point> > contours;
vector<Vec4i> hierarchy;
Mat img_Vehic;
imgBin.copyTo(img_Vehic );

```

```

findContours(img_Vehic,contours,hierarchy,CV_RETR_EXTERNAL,CV_CH
AIN_APPROX_SIMPLE);
CvRect rectP[1000];
int centro_Y, centro_X, centroAnt_Y = 0, centroAnt_X = 0, direcc_Y,
direcc_X;
for( int i = 0; i< contours.size(); i++ )
{
    double a=contourArea(contours[i],false);
    if ( a > 3000 && a < 50000 )
    {
        rectP[i] = boundingRect(contours[i]);
        centro_X = int(rectP[i].x + (rectP[i].width/2));
        centro_Y = int(rectP[i].y + (rectP[i].height/2));
        if (centro_X > 300 && centro_Y > 100 && centro_Y <
350)
        {
            rectangle(frame1, rectP[i], Scalar(0,255,0), 2);

```

```

        direcc_X = abs(centro_X - centroAnt_X); //para
saber si es el mismo vehiculo

        direcc_Y = centro_Y - centroAnt_Y;

//circle(frame1,Point(centro_X,centro_Y),8,Scalar(255,0,255),5);

        //if (direcc_1 < 0 && abs(direcc_1) < 50 &&
centro_Y > 200 && centro_Y < 250) // Hacia arriba
        if (centro_Y > 250 && centro_Y < 265)
        {

circle(frame1,Point(centro_X,centro_Y),8,Scalar(255,0,255),5);

        //circle(frame1,Point(400,50),10,Scalar(0,0,255),10);

        //if (centro_X > centroAnt_X-100 &&
centro_X < centroAnt_X+100)

        //{

                car++;

                putText(frame1, to_string(car),
Point(rectP[i].x, rectP[i].y-20), FONT_ITALIC,1.2,Scalar(0,0,255),2);

        //}

        }

        }

        putText(frame1, "Cant: " + to_string(car), Point(500, 50),
FONT_ITALIC,1.2,Scalar(0,255,0),3);

        centroAnt_Y = centro_Y;
        centroAnt_X = centro_X;

    }

}

```



```
resizeWindow("Video",640,480);
imshow("Video",frame1);

frame2.copyTo(frame1);

//if (cap.get(CV_CAP_PROP_FRAME_COUNT) < 2) break;
if(waitKey(10) == 27) break;    // ESC
}
return 0;
}
```